

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínské inženýrství

Online vizualizace provozních údajů a polohy
vozidla

On-line visualization of traffic data and vehicle position

2014

Ondřej Rychtar

Zadání bakalářské práce

Student: **Ondřej Rychtar**
Studijní program: B2649 Elektrotechnika
Studijní obor: 2601R004 Měřicí a řídicí technika
Téma: **Online vizualizace provozních údajů a polohy vozidla - BC**
On-line Visualization of Traffic Data and Vehicle Position

Zásady pro vypracování:

1. Seznámení se s geografickými informačními systémy.
2. Seznámení se s problematikou telemetrie vozidel.
3. Seznámení se s databázovými prostředky (SQL).
4. Navržení řešení pro zpracování a vizualizaci telemetrických dat nad mapovým podkladem.
5. Experimentální ověření navrženého řešení.
6. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] THURSTON, Jeff, Thomas K. POIKER and Patrick J. MOORE. *Integrated geospatial technologies: a guide to GPS, GIS, and data logging*. Hoboken, N.J.: John Wiley, c2003, xiii, 266 p. ISBN 0471244090.
- [2] HOJGR, Radek a Jan STANKOVIČ. *GPS: praktická uživatelská příručka*. 1. vydání. Brno: Computer Press, a.s., 2007. 221 s. ISBN 978-80-251-1734-7.
- [3] BROWN, Martin C. *Hacking Google Maps and Google Earth*. Indianapolis, IN: Wiley Pub., c2006, xxiii, 373 p. ISBN 978-0471790099.
- [4] KNOX, Bob. *A practical guide to race car data analysis*. Rev. 1 ed. [S.l.: CreateSpace], 2010. ISBN 978-1456587918.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. David Vala**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



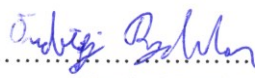
prof. RNDr. Václav Snášel, CSc.
děkan fakulty



Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 7. 5. 2014

Podpis studenta.....
Ondřej Rychtar

Poděkování

Děkuji vedoucímu mé bakalářské práce, panu Ing. Davidu Valovi za hodnotné rady a odborné vedení během práce. Dále bych chtěl poděkovat panu Ing. Zdeňku Slaninovi, Ph.D, panu Ing. Vítovi Otevřelovi a panu Ing. Jaromíru Konečnému za užitečné rady při tvorbě mé bakalářské práce.

Abstrakt

Tato bakalářská práce se zabývá online vizualizace provozních údajů a polohy vozidla. V praktické části je úkolem vytvořit aplikaci a databázi a v teoretické části je úkolem zpracovat informace o geografických informačních systémech, databázových prostředcích SQL a telemetrii. Aplikace bude zobrazovat data uložené ve vytvořené databázi na serveru. Aplikace je vytvořena pomocí softwarového nástroje Microsoft Visual Studio. V této aplikaci je možné volit mezi několika vozidly. Poloha vozidla, trasa a další informace budou zobrazovány na mapě. Dále jsou v aplikaci pro lepší vizualizaci grafy, analogové ukazatele a stavový řádek.

Klíčová slova

C#, MySQL, GMap.NET, GPS

Abstract

This thesis deals with online visualization of operating data and vehicle position. In the practical part of the task is to create an application and a database in the theoretical part of the task to process information about geographic information systems, database SQL and telemetry devices. The application will display the data stored in the database created on the server. The application is created using software tools Microsoft Visual Studio. In the application it is possible to choose among several vehicles. Location of vehicle, route and other information will be displayed on the map. Furthermore the application for better visualization of graphs, analog indicators and status bar.

Key words

C#, MySQL, GMap.NET, GPS

Seznam použitých zkratek

AuC	z angl. Authentication Centre
BCNF	Boyce-Coddova normální forma
BS	z angl. Base Station
BSC	z angl. Base Station Controller
BTS	z angl. Base Transceiver Station
C#	z angl. C Sharp
CAD	z angl. Computer-aided design
CAN	z angl. Controller Area Network a BUS
DDL	z angl. Data Definition Language
DML	z angl. Data Manipulation Language
EIR	z angl. Equipment Identity Register
ESA	z angl. European Space Agency
EU	Evropská unie
GIS	Geografický informační systém
GLONASS	z rus. Globalnaya navigatsionnaya sputnikovaya sistema
GMap.NET	z angl. Great Maps for Windows Forms & Presentation
GMS	z angl. Global System for Mobile Communications
GNSS	z angl. Global Navigation Satellite System
GPRS	z angl. General packet radio service
GPS	z angl. Global Positioning System
HLR	z angl. Home Location Register
MSC	z angl. Mobile Services Switching Centre
MySQL	z angl. My Structured Query Language
SAR	z angl. Search and Rescue
SDL	z angl. Storage Definition Language
S-JTSK	Systém jednotné trigonometrické sítě katastrální
SQL	z angl. Structured Query Language
TCP	z angl. Transmission Control Protocol
TCP/IP	z angl. Transmission Control Protocol/Internet Protocol
UDP	z angl. User Datagram Protocol
UTM	z angl. Universal Transverse Mercator Geographic Coordinate System
VDL	z angl. View Definition Language
VLR	z angl. Visitor Location Register
WGS	z angl. World Geodetics System

Obsah

1	Úvod	1
2	GIS.....	2
2.1	Souřadnicové systémy.....	3
2.1.1	WGS 84.....	3
2.1.2	UTM	4
2.2	Navigační systémy	5
2.2.1	GPS.....	6
2.2.2	GLONASS	6
2.2.3	Galileo.....	6
2.2.4	COMPASS.....	7
3	Telemetrie vozidel	8
3.1	Poloha vozidla.....	8
3.2	Teplota vnějšího prostředí.....	8
3.3	Akcelerometrie.....	9
3.4	GMS a GPRS.....	10
3.4.1	GMS.....	10
3.4.2	GPRS	11
3.5	CAN BUS	13
4	Databáze a jazyk SQL.....	14
4.1	Relační databáze	14
4.1.1	Základní pojmy	14
4.1.2	Normální formy.....	15
4.2	Jazyk SQL.....	17
5	Vlastní návrh aplikace.....	18
5.1	Měřicí a vizualizační řetězec.....	18
6	Softwarové řešení	19
6.1	Struktura databáze.....	19

6.2	Vizualizační aplikace	21
7	Testování aplikace	29
8	Závěr	31
	Literatura	33
	Seznam příloh	35
	Příloha I. Vizualizace dat na hybridních mapách od Google.....	36
	Příloha II. Vizualizace dat na mapách od Mapy.cz	37

1 Úvod

Vizualizace v dnešní době se nachází v mnoha oblastech. Ale hlavně v průmyslu. A zejména v těchto odvětvích v automobilovém, strojírenském, energetickém, stavebním a také v medicíně, geografii apod. Setkat se s nimi můžeme v různě velkých provozech, jako jsou pivovary, pekárny atd.

Pomocí vizualizace jde sledovat a řídit procesy v reálném čase. Dále lze ovládat procesy na dálku a to pomocí internetu, intranetu, atd. Ve vizualizaci je možné zobrazovat různé alarmové stavy a hlášení. Pomocí vizualizace lze ukládat a načítat měřená data. Má široké využití z důvodu snadné a jednoduché obsluhy. Ve vizualizaci jdou nastavit přístupová práva pro jednotlivé uživatele. A tím umožnit využívání některých funkcí jen určeným uživatelům.

Cílem této práce je vytvořit aplikaci, která bude online vizualizovat provozní údaje a polohu vozidla. Data získané z automobilu jsou ukládány v určitém časovém intervalu do databáze na serveru. Odkud budou čtena a zobrazována. Jeden z hlavních parametrů vizualizace je poloha vozidla. Ta se bude zobrazovat na mapě. Kde se budou zobrazeny i další údaje, jako je rychlost, zrychlení, atd. Informace o zvoleném vozidle budou zobrazovány v tabulkách.

Z těchto údajů se dá vozidlo sledovat v reálném čase, kde se nachází a jakou jede rychlostí. A to aniž by uživatel musel být ve vozidle. Aplikace se dá využít pro bezpečnostní účely a to tak že se dané vozidlo dá sledovat, zda nebylo odcizeno. Nebo pro firmy na kontrolu služebních aut. Data jsou zobrazovány tak aby jim porozuměl i laik.

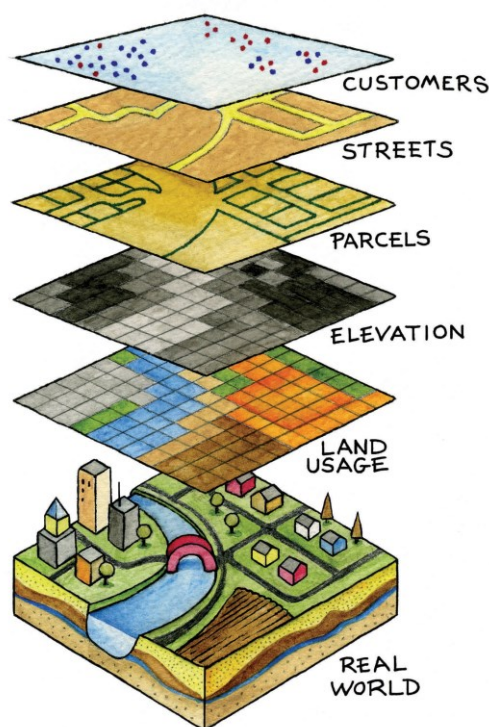
Pro určení polohy se v dnešní době využívají globální navigační systémy. Ale málo lidí ví, jak tyto systémy pracují. Že existují zařízení, která spojují několik navigačních systémů dohromady pro lepší přesnost a spolehlivost. Většina lidí zná jen jeden navigační systém a to je americký GPS. Jak jsem již zmínil, existuje kombinace amerického GPS a ruského GLONASS. Tato kombinace je tak rozšířená, že se nachází již v základním vybavení většiny chytrých mobilních telefonů.

Mezi nejrozšířenější bezdrátové přenosy patří technologie GPRS a GSM. Tyto technologie využíváme každý den. A většina z nás ani netuší, co je potřeba pro uskutečnění hovoru nebo připojení k mobilnímu internetu pomocí mobilního telefonu. A s čím se musí vypořádat poskytovatelé těchto služeb při narůstajícím počtu zařízení.

Dále se bakalářská práce zabývá základy databází a jazyka SQL. Jsou zde základní pravidla pro tvorbu databází. Princip a použití jazyka SQL. Ukázky z webového nástroje phpMyAdmin. Pro praktickou část práce byla použita databáze MySQL. Z důvodu jednoduché správy přes phpMyAdmin. A své další výhody, které jsou uvedeny níže.

2 GIS

GIS (Geografický informační systém) je elektronický systém zabývající se zpracování geografických informací. Geografická informace je celkový údaj o hmotném nebo nehmotném objektu, který musí obsahovat údaj o geografické poloze objektu. To znamená, že je potřeba uchovávat souřadnice objektu. Nehmotný objekt může být údaj o daném stromě nacházející se na určitých souřadnicích. Systém je navržen tak, aby mohl manipulovat, analyzovat, zpracovávat geografické informace. GIS je poměrně široký pojem, který obsahuje množství různých technologií, postupů a metod. Má mnoho uplatnění v inženýrství, plánování, řízení, dopravě / logistice, pojišťovnictví, telekomunikaci a obchodu. Moderní technologie GIS využívají digitální informace, které používají různé metody vytváření digitálních dat. Nejběžnější způsob vytvoření údajů je digitalizace, kde papírová mapa nebo průzkumný plán se převádí do digitální podoby pomocí programu CAD. Díky široké dostupnosti snímků (a to jak z družicových a leteckých zdrojů), digitalizace se stává hlavním prostředkem, jehož prostřednictvím jsou geografické údaje extrahovány. ^{[8] [16]}



Obrázek 1 GIS ve vrstvách. ^[17]

2.1 Souřadnicové systémy

Souřadnicový systém obecně slouží k popisu polohy bodu v prostoru. V dnešní době se využívá několik souřadnicových systémů. Rozlišuje dva druhy systémů. Globální a lokální.

Globální jsou tyto:

- WGS 84 – World Geodetics System
- UTM – Universal Transverse Mercator Geographic Coordinate System

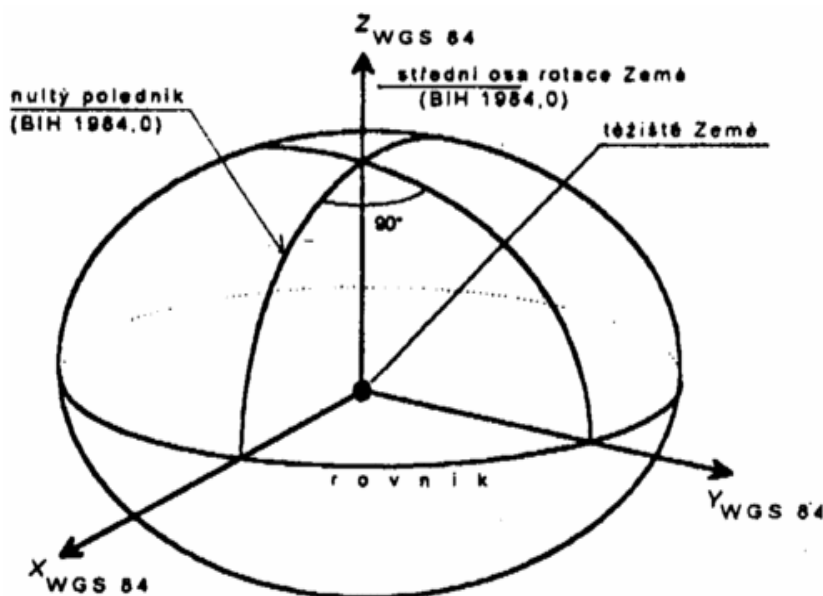
Lokální systémy:

- S 42
- S-JTSK - systém jednotné trigonometrické sítě katastrální

Pro navigační systémy se používají globální souřadnicové systémy. [8]

2.1.1 WGS 84

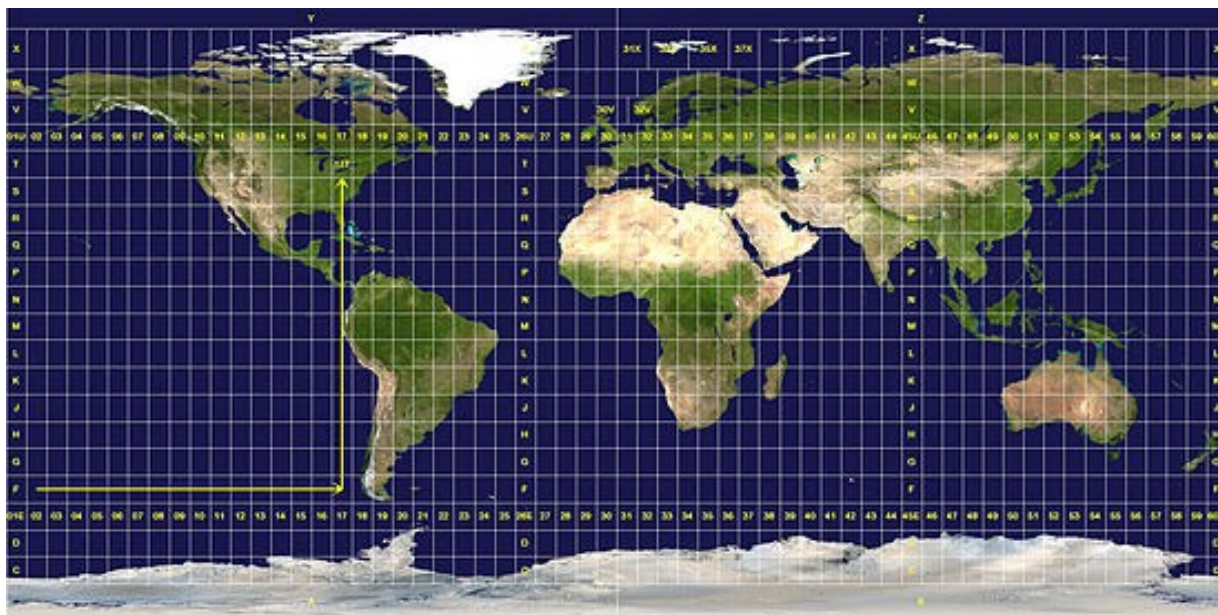
Používány v mnoha oborech zajímavící se o pohyb na Zemi. Byl vydán ministerstvem obrany Spojených států amerických v roce 1984. Je to tradiční souřadnicový systém šířky-délky. Zeměpisná šířka bodu je úhel, který svírá přímka vedena tímto bodem do středu Země s rovinou rovníku. Rovník rozděluje Zemi na dvě polokoule – severní a jižní. Zeměpisná šířka severní polokoule se udává s kladným znaménkem a jižní polokoule se záporným. To znamená, že na pólech je zeměpisná šířka $\pm 90^\circ$. Zeměpisná délka je úhel polední procházející bodem ke Greenwichskému poledníku. Rozlišujeme dvě zeměpisné délky – východní a západní. Východní délka se udává se záporným znaménkem. A západní s kladným. [8] [9]



Obrázek 2 Souřadnicový systém WGS-84 [9]

2.1.2 UTM

Je to souřadnicový systém, při kterém se Země rozdělí po polednících na poledníkové pásy. A tyto pásy se dále rozdělí na pásy zeměpisné šířky. Poledníkové pásy se značí čísla a pásy zeměpisné šířky písmeny. Tak je Země rozdělena do zón jako šachovnice. Vzdálenost dvou bodů se dá vypočítat pomocí Pythagorovy věty, ale oba body se musí nacházet ve stejné zóně. Souřadnicový systém je v metrech. Každá zóna má jiný střed souřadnic. Střed je tvořen průsečíkem středu poledníkového pásu s rovníkem. [8] [11]

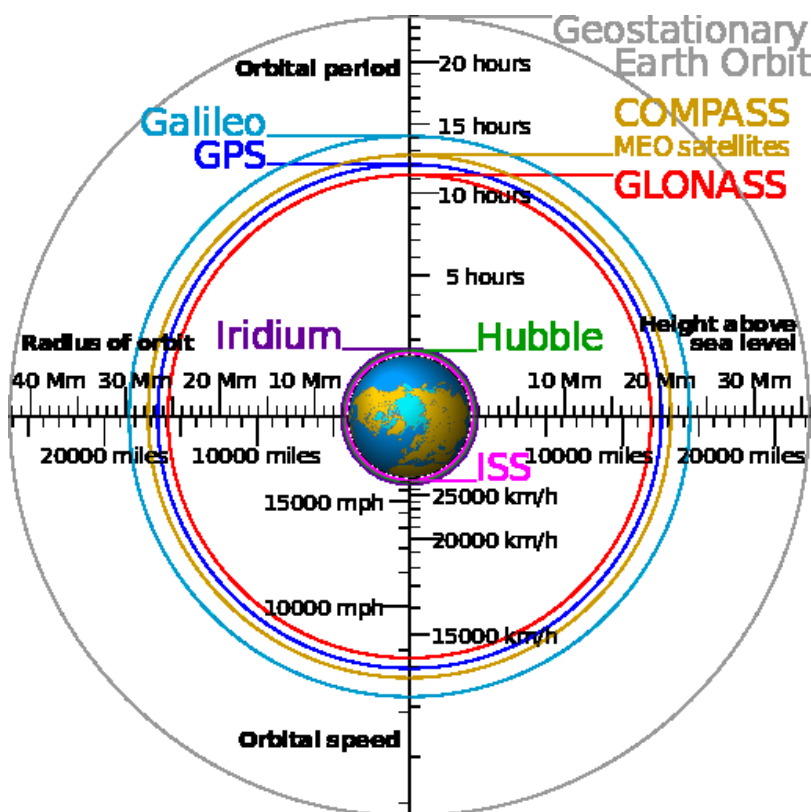


Obrázek 3 Souřadnicový systém UTM [11]

2.2 Navigační systémy

V této kapitole se budeme zabývat navigačními systémy:

- GPS - Globální navigační systém USA. Plně funkční po celém světě.
- GLONASS - Globální navigační systém Ruska. Plně funkční po celém světě.
- Galileo - Globální navigační systém vyvinut Evropskou unií a dalšími partnerskými zeměmi, plánované uvedení do provozu od roku 2014 (a plně funkční od roku 2019).
- Beidou - Regionální navigační systém v Číně, v současné době jen pro Asii a západní Pacifik.
- COMPASS – Globální navigační systém v Číně, plánované uvedení do provozu od roku 2020.
- IRNSS - Regionální navigační systém v Indii, plánované uvedení do provozu od roku 2014. Pokrytí jen pro Indii a severní Indický oceán.
- QZSS - Japonský regionální navigační systém pokrývající Asii a Oceánii.^[1]



Obrázek 4 Navigační systémy kolem Země ^[1]

2.2.1 GPS

GPS (Global Positioning System) je navigační satelitní systém pro určení polohy v reálném čase kdekoli na Zemi nebo v její blízkosti. Je spravován vládou Spojených států amerických. Tento systém je využíván pro civilní, vojenské a obchodní potřeby po celém světě. A je volně přístupný každému GPS přijímači.

Navigační systém je tvořen v současné době 32 družic, které obíhají Zemi ve výšce 20,2 km. Pro určení polohy přijímače je zapotřebí minimálně 4 družic. Uvnitř každé družice jsou velmi přesné atomové hodiny. GPS čas je teoreticky přesný 14 nanosekund. Nicméně, většina přijímačů způsobuje ztrátu přesnosti při interpretaci signálů a jsou přesné jen na 100 nanosekund. Po přijetí signálu z několika satelitů je možné z rozdílných časů a z doby šíření signálů a polohy družic, dopočítat polohu přijímače.

- Civilní využití – Všechny civilní GPS přijímače mají omezení kvůli bezpečnosti.
- Vojenské využití – Je pro navigování jednotek, řízených stíhacích, bezpilotních letounů, atd.^{[1] [5] [6]}

2.2.2 GLONASS

GLONASS (Rusky : ГЛОНАСС) je zkratkou Globalnaya navigatsionnaya sputnikovaya sistema což znamená globální družicový navigační systém. Je to podobný systém jako GPS a je to jedinou alternativou navigační systém v provozu s celosvětovým pokrytím a srovnatelné přesnosti. GLONASS poskytuje v reálném čase polohu a určení rychlosti pro vojenské a civilní uživatele. Družice jsou umístěny ve střední kruhové dráze ve výšce 19,1 km. GLONASS se skládá z 24 družic, z toho je 18 pro určení polohy v Rusku. Minimální počet družic pro určení polohy je 4.

Systém používá dva druhy signálu:

- Civilní signál – Je omezený z důvodu bezpečnosti.
- Armádní signál – Slouží pro armádní potřeby.

V současné době se také využívá kombinace GPS a GLONASS systému. A to umožňuje rychlou odezvu ve velkých městech nebo v horách. Tento systém využívá více než 50 družic.^[2]

2.2.3 Galileo

Galileo je globální družicový navigační systém (GNSS), v současné době stavěn Evropskou unií (EU) a Evropskou kosmickou agenturou (ESA). První určení polohy pouze ze satelitů systému Galileo bylo dosaženo dne 12. března 2013. Úplné dokončení třiceti satelitů systému Galileo (27 pro provoz a tři náhradních) se očekává v roce 2019. Satelity budou ve výšce 23,222 km. A budou na třech oběžných drahách. Na každou oběžnou dráhu připadne 9 aktivních satelitů a jeden náhradní.

Systém Galileo bude mít pět hlavních služeb:

- Volný přístup k navigaci - K dispozici bez poplatku pro každého, přesnost až 1 metr.

- Obchodní navigace (šifrovaná) - Vysoká přesnost na centimetry, za kterou poskytovatelé budou účtovat poplatky.
- Bezpečná navigace – Volný přístup pro aplikace, kde je nezbytné zaručit přesnost. Zprávy o chybách v systému.
- Veřejná regulovaná navigace (šifrovaná) - Nepřetržitá dostupnost i když ostatní služby jsou zakázány v době krize, vládní agentury budou hlavními uživateli.
- Vyhledávání a záchrana - Bude poskytovat jedinečnou globální „Funkci hledání a záchrany“ (SAR). Družice bude vybavena transpondérem, který bude předávat tísňový signál z vysílače uživatele do záchranného koordinačního centra, které bude moci následně zahájit záchrannou operaci. Ve stejnou dobu, bude systém poskytovat signál pro uživatele, že jeho situace byla oznámena, a že pomoc je na cestě. ^{[3] [7]}

2.2.4 COMPASS

COMPASS nebo Beidou-2, je globální družicový navigační systém, který se bude skládat z 35 družic, je ve výstavbě od ledna 2013. Podle plánu, začne sloužit zákazníkům po celém světě po dokončení v roce 2020.

K dispozici budou dvě úrovně poskytovaných služeb. Bezplatná služba pro civilisty a licencovaných služeb pro čínskou vládu a armádu. Pro civilní služby bude přesnost polohy 10 metrů, hodiny s přesností na 10 nanosekund, a přesností rychlosti už od 0.2 m/s. Omezená vojenská služba má mít přesnost 10 cm. Mohou být využity pro komunikaci, a poskytovat informace o stavu systému uživateli. ^[4]

3 Telemetrie vozidel

Telemetrie je automatizovaný proces měření, přenosu a shromažďování dat z míst, kde by měření bylo obtížné nebo nebezpečné. Slovo je odvozeno z řeckých slov *tele* = vzdálené a *metron* což je měření. Mnoho telemetrických systémů využívá k přenosu dat GSM / GPRS síť. Díky této technologii se dají data šířit pomocí internetu po celém světě.^[10]



Obrázek 5 Telemetrie vozidel^[15]

Telemetrie vozidel se zabývá měřením dat pomocí senzorů umístěných jak venku tak i uvnitř vozidla a přenosu pro další zpracování. Data se ukládají obvykle do databáze pro lepší přehlednost. Nejčastější přenášená data jsou rychlost, zrychlení, vzdálenost a poloha.

3.1 Poloha vozidla

Z údajů o poloze vozidla se dá získat mnoho dat. Jako je rychlost, ujetá vzdálenost a doba jízdy. Pro určení polohy vozidla se používají globální navigační systémy. A to pomocí zařízení, které je umístěné ve vozidle a přímá signál z družic kolem Země. V některých oblastech jako třeba lesy nebo velkoměsta může dojít ke ztrátě signálu z družic. Aby k tomu nedocházelo, existují systémy, které kombinují více navigačních systémů. Nejčastější kombinací je ruský GLONASS a americký GPS. Zařízení pro určení polohy může být nainstalováno napevno, jako výbava automobilu nebo přenosné externí navigace, či mobilní telefon.

3.2 Teplota vnějšího prostředí

Je to údaj, který slouží k určení, v jakém prostředí se automobil nachází. Obvykle je získán z čidla umístěného výrobcem automobilu. Čidlo je umístěno tak aby nedocházelo k ovlivňování měřené teploty motorem nebo jízdou. Nejčastěji se používají pro měření teploty odporové snímače. Odporový snímač funguje na principu, když se zvýší teplota, zvýší se odpor snímače. Na odporovém snímači se měří napětí a proud. A ze získaných údajů je pomocí Ohmova zákona vypočten odpor snímače. Podle velikosti odporu se vyhledá v tabulce příslušná teplota.

3.3 Akcelerometrie

Je měření zrychlení pomocí senzorů akcelerometrů. Zrychlení je změna rychlosti hmotného bodu nebo soustavy hmotných bodů v čase. Zrychlení je vektorová veličina, kterou definujeme takto:

$$a = a_n + a_t = dv/dt, \text{ kde} \quad (1)$$

a – vektor okamžitého zrychlení [ms^{-2}]

a_n – vektor normálové složky zrychlení [ms^{-2}]

a_t – vektor tečné složky zrychlení [ms^{-2}]

v – vektor rychlosti [ms^{-1}]

d/dt – časová derivace

Podle hodnot normálového a tečného zrychlení se dá učit, o jaký pohyb se jedná:

- Rovnoměrný přímočarý $a_t = 0$, $a_n = 0$
- Nerovnoměrný přímočarý $a_t \neq 0$, $a_n = 0$
- Rovnoměrný křivočarý $a_t = 0$, $a_n \neq 0$
- Nerovnoměrný křivočarý $a_t \neq 0$, $a_n \neq 0$

Pro měření zrychlení se používá:

1. Druhý Newtonův zákon (zákon síly):

$$F = m \cdot a, \text{ kde} \quad (2)$$

F – vektor síly [N]

m – hmotnost [kg]

a – vektor zrychlení [ms^{-2}]

V případě, že síla F je způsobena gravitací Země, pak zrychlení nazýváme tíhové a značíme g .

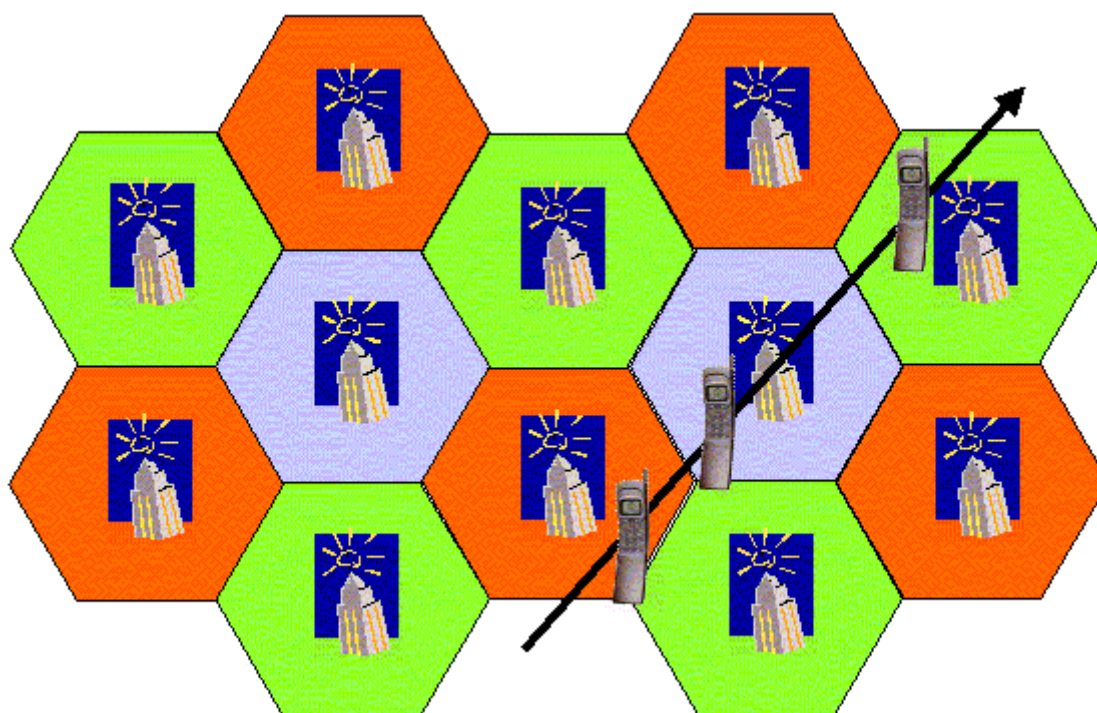
2. Třetí Newtonův zákon (zákon akce a reakce): při vzájemném působení hmotných útvarů, prostředí, vznikají vždy vzájemná silová působení, která jsou stejně veliká, ale mají vzájemně opačný směr (působí proti sobě). ^[19]

3.4 GMS a GPRS

Tato kapitola se bude zabývat technologií sítí GMS a GPRS. Jejich základními parametry.

3.4.1 GMS

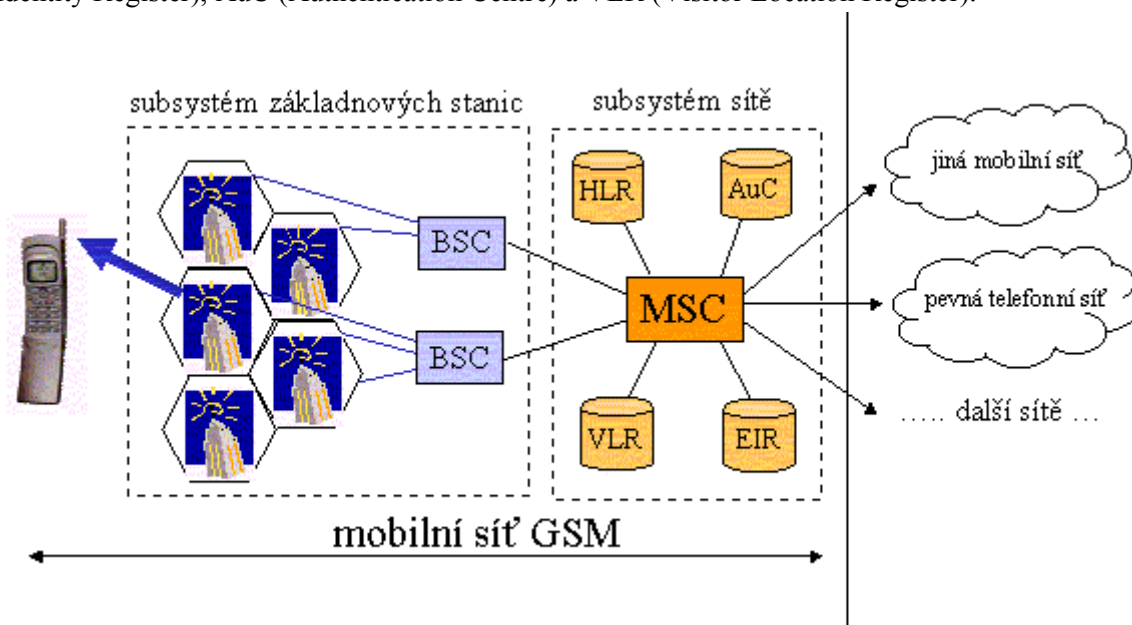
Zkratka GMS vychází z těchto slov Global System for Mobile Communications. V překladu to je globální systém mobilní komunikace. Mobilní síť ke svojí komunikaci využívá rádiových vln. Frekvence pro mobilní síť jsou pevně určené a každý operátor jich dostává jen několik od státního orgánu pověřený správou frekvenčního spektra. V České Republice je to Český telekomunikační úřad. Aby operátor mohl přidělit každému probíhajícímu hovoru samostatný komunikační kanál. Musí při nedostatku volných frekvencí využívat některé frekvence pro více různých hovorů najednou. Je zapotřebí zajistit aby se hovory neovlivňovali navzájem. A toho se dosáhne pomocí buňkového principu. Jeho principem je rozdělení určitého území, na kterém operátor provozuje své služby. Na vhodně velké území (buňky), uspořádané tak aby se v sousedních buňkách využívaly jiné frekvence. V praxi se využívají buňky o tvaru šestiúhelníku. Proto stačí použít tři druhy buněk. Toto uspořádání je omezeno počtem frekvencí, které má daná buňka k dispozici. Pokud je potřeba navýšení souběžných hovorů, pak se musí zhustit síť buněk.



Obrázek 6 Síť buněk ^[12]

V každé buňce v jejím středu se nachází BS (Base Station) nebo BTS (Base Transceiver Station). Má za úkol komunikovat s terminály, které se nachází uvnitř příslušné buňky. A to na frekvencích, které byly určeny buňce. Pokud zařízení přejde z jedné buňky do druhé, stanice to poznají a komunikaci si předají. Stanice musí být navzájem propojeny a společně řízeny. V praxi se obvykle řídí několik buněk najednou a to pomocí Base Station Controller (BSC). Všechny základní stanice jsou připojeny k

centrální ústředně (MSC, Mobile Services Switching Centre). Lze si ji představit jako klasickou analogovou ústřednu z pevné sítě. Slouží ke směrování hovorů k příjemcům. Z důvodu pohybu mobilních zařízení musí být vedena databáze (HLR, Home Location Register), kde se zařízení právě nalézá. A dále zde musí být vedena evidence zařízení z jiných států a k tomu slouží EIR (Equipment Identity Register), AuC (Authentication Centre) a VLR (Visitor Location Register). ^{[12][13]}



Obrázek 7 Architektura sítě GSM ^[12]

Síť GSM nabízí několik služeb. Podle využití sítě. Jsou to tyto:

- Telematické služby – Jedná se o služby, které využívají lidé přímo. Např.: telefonní hovor.
- Nadstavbové služby- Jde o transportní služby. K využívání je potřeba doplňující zařízení nebo software. Např.: Internet.
- Doplňkové služby – Jedná se o funkce, které jsou k dispozici v síti. Ale mobilní telefon je musí umět využívat. Typický příklad je přijetí druhého hovoru.
- Funkce SIM karty – SIM karta umožňuje ukládat data uživatele mobilního telefonu. Ale také poskytuje vlastní služby jako je například SIM Toolkit.

Pásmo provozu mobilních telefonů (GSM) v České republice je 870–960 MHz. ^[13]

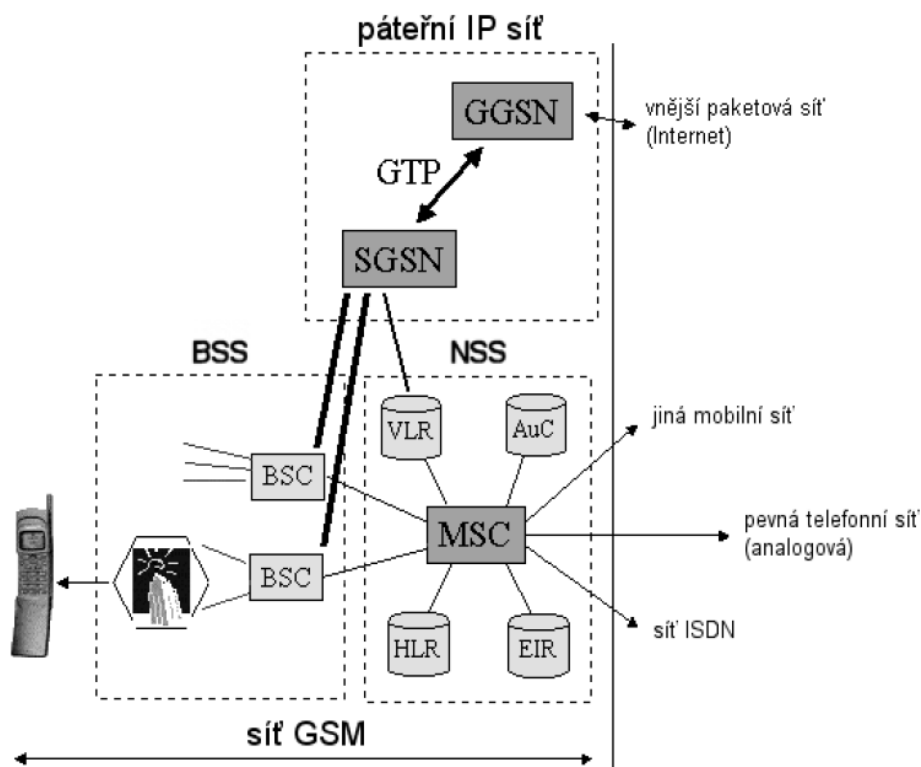
3.4.2 GPRS

Technologie GPRS (general packet radio service) je zásadní změnou datových přenosů. Umožňuje vznik zcela nových aplikací a služeb, které se nedali zavádět kvůli ekonomické efektivnosti. Z toho důvodu se musely zvést větší změny v síti GSM. Díky těmto změnám bylo zavádění GPRS nákladnější záležitostí. A přesto je to velmi rozšířená technologie. V České republice ji využívají všichni operátoři. Velkou odlišností GPRS od přenosu dat pomocí mobilní sítě GSM je, že předchozí přenosy fungovaly na principu přepojování okruhů. GPRS funguje na principu přepojování paketu. To má obrovskou výhodu v tom, že uživatelé využívají síť jen tehdy, když přenášejí data. Tím neblokují síť po celou dobu spojení.

Zavedení GPRS si vyžádalo úpravu sítě GSM a to tak že byly přidány nové prvky do stávající sítě. Sít' byla rozšířena o tyto dva prvky:

- uzly SGSN (serving GPRS support node)
- uzly GGSN (gateway GPRS support node)

Uzly je v síti použito podle toho, kolik je jich zapotřebí. Pro každou vnější datovou síť musí existovat samostatný uzel GGSN. Uzly SGSN a GGSN mezi sebou komunikují pomocí protokolu GTP (GPRS tunneling protocol). Což je protokol ze skupiny TCP/IP. Využívá ke svému fungování transportní protokoly UDP nebo TCP.



Obrázek 8 rozšíření sítě GSM o GPRS ^[13]

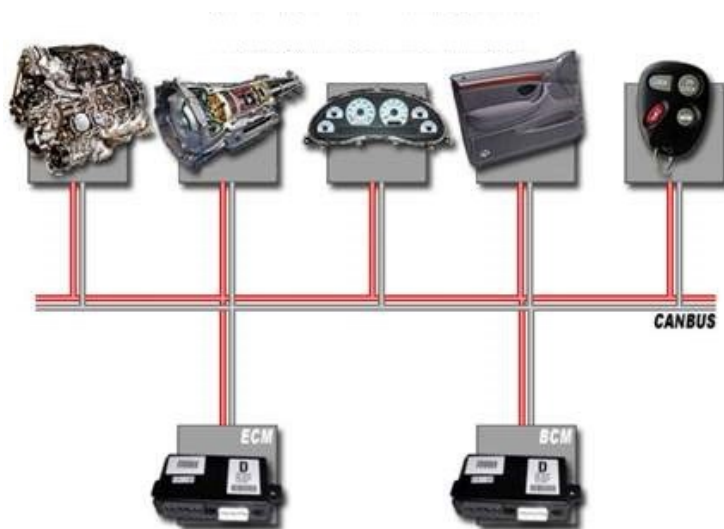
Princip přenosu dat je realizován v několika variantách:

- Nespojovaný (connectionless) přenos – Pakety jsou přenášeny nezávisle na sobě. Mezi příjemcem a odesílatelem není navázáno spojení. Pakety jsou posílány různými trasami a v různém pořadí. Pracuje na podobném principu jako TCP/IP.
- Spojovaný (connection-oriented) přenos – Mezi příjemcem a odesílatelem je navázáno spojení, ale jen pro vytyčení cesty, jakou budou pakety cestovat. U tohoto přenosu jsou pakety posílány vytyčenou cestou a v určeném pořadí.

Data před přenosem musí být rozporcována na pakety. A tyto pakety jsou celé přenášeny k příjemci. Přenosové kapacity jsou vždy k dispozici a nejsou trvale někomu přiřazovány. To umožňuje využívat celé kapacity k odeslání nebo příjmu dat. ^[13]

3.5 CAN BUS

Zkratka znamená CAN – Controller Area Network BUS – sběrnice. Z důvodu narůstající kabeláže v automobilu byla zavedena sběrnice s jedním párem kabelů. Tato sběrnice umožňuje komunikaci všech elektronických jednotek navzájem. CAN BUS je datová sběrnice. Její maximální rychlost může být až 1 Mbit/s. Sběrnice byla vyvinuta roku 1983 firmou Robert Bosch GmbH. První auto na trhu s touto sběrnicí bylo BMW 850 coupe v roce 1986. CAN BUS se stalo všeobecným standardem. Od roku 2008 musí být automobily prodávané v severní Americe a také v Evropské unii vybaveny touto technologií. CAN BUS umožňuje z jediného místa v automobilu zjistit informace o otevření dveří nebo kufru, o zapnutí zapalování, rozsvícení světel, rychlosti auta, aktivaci brzdy a mnoho dalších informací. Komunikace elektronických jednotek se liší nejen podle značky auta, ale také podle modelu nebo roku výroby. Převodníky pro komunikaci na CAN BUS ovládají komunikaci pro několik desítek až stovek aut. Výrobci těchto převodníků se snaží tento seznam stále rozšiřovat a doplňovat o nová auta na trhu. ^[14]



Obrázek 9 Kabeláž CAN BUS ^[14]

Výhody:

- Vysoká přenosová rychlost dat 1Mbit/s při délce sběrnice do 40 m
- Prioritní přístup zabezpečující urychlené doručení významných zpráv
- Diagnostika sběrnice
- Vysoká úroveň zabezpečení přenosu
- Vysoká provozní spolehlivost
- Široká dostupnost součástek
- Nízká cena

Nevýhody:

- Omezený počet dat přenesených v rámci jedné zprávy
- Prvotní náročnost nastavení registrů CAN sběrnice

4 Databáze a jazyk SQL

Databáze je soubor dat popisující reálný svět (Evidence občanů, Seznam žáků). Entita je jedna část reálného světa (stroj, člověk, student, kniha), který má určité vlastnosti. Vlastnosti popisující entitu nazýváme atribut (jméno, příjmení, věk, plat). Jelikož entity odpovídají prvkům z reálného světa, mají mezi sebou určitý vztah. Existuje několik vazeb mezi entitami. První je vazba 1:1 to znamená, že jedné entitě odpovídá jiná entita. Např.: Jedna objednávka může být jen jednou zaplacená. Druhý typem vazby je vazba 1:N, je to vazba kdy pro jeden záznam odpovídá někde jinde N jiných záznamů. Např.: Jeden zákazník může mít několik objednávek. Posledním typem vazeb je typ N:M. Tento typ vazby nemá žádné omezení. Např.: Student na vysoké škole se může zapsat na několik předmětů a zároveň na jeden předmět se může zapsat několik studentů. Databázový model, tento pojem byl zaveden matematiky. Jedná se o prostředek popisující databázi. Používaly se dva druhy databázového modelu. První hierarchický založen na principu hierarchie mezi entitami se vztahy podřízenosti a nadřazenosti. Druhým tipem byl síťový, vycházel z teorie grafů, kde uzly v grafu odpovídají entitám a orientované hrany definují vztahy mezi entitami. V 70. letech se ukázaly tyto modely jako nedostačující a proto vznikl relační model, který je standardem dodnes. ^[18]

4.1 Relační databáze

Základní pojem je relace. Lze si ji představit jako tabulku s řádky a sloupcečky. Kde sloupce jsou vlastnosti (atributy) entit a údaje v řádku jsou informace o aktuálním stavu.

ID	PŘÍJMENÍ	JMÉNO	ČÍSLO_RP
1	novák	jan	234563
2	nový	petr	326514
3	nováček	jan	235517

Obrázek 10 Tabulka databáze

Tabulka je základním prvkem databáze. Jeden řádek v tabulce je databázový záznam. Soubor tabulek tvoří databázi (relační schéma). Tabulka nám popisuje entitu. Za sloupce si volíme atributy, které k dané entitě chceme evidovat.

4.1.1 Základní pojmy

Uživatelská data jsou hodnotou. Každý sloupec v tabulce má vlastní datový typ. Jeden sloupec se obvykle volí pro primární klíč. To je jedinečné číslo pro každý řádek tabulky. Dále je zde funkční závislost, to je závislost dvou atributů. Např.: Velikost platu podle vykonané práce nebo velikost náhrady jízdného podle délky trasy.

4.1.2 Normální formy

Normální formy se používají pro správný návrh databáze. Jsou čtyři.

1. První norma (1NF) říká, že všechny atributy jsou atomické. Tzn. Dále nedělitelné. Například adresa, bude obsahovat následující atributy: Jméno, příjmení a bydliště.

JMÉNO PŘÍJMENÍ		BYDLIŠTĚ
jan	novák	Ostravská 16, Praha 16000
petr	nový	Svitavská 8, Brno 61400
jan	nováček	Na bradlech 1147, Ostrava 79002

Obrázek 11 Tabulka bez 1NF ^[18]

Kdybychom z této tabulky chtěli vypsát lidi se stejným směrovacím číslem. Tak by se nám to nepovedlo. Protože atribut BYDLIŠTĚ není atomický. Proto musíme BYDLIŠTĚ rozdělit na ulici, číslo, město a PSČ.

JMÉNO	PŘÍJMENÍ	ULICE	ČÍSLO	MĚSTO	PSČ
jan	novák	Ostravská	16	Praha	16000
petr	nový	Svitavská	8	Brno	61400
jan	nováček	Na bradlech	1147	Ostrava	79002

Obrázek 12 Tabulka podle 1NF ^[18]

2. Druhá normální forma (2NF) – tabulka musí splňovat 1NF a dále každý atribut musí být závislý na primárním klíči nebo být primárním klíčem.

ČÍSLO	JMÉNO	PŘÍJMENÍ	ČÍS_PRAC	NÁZEV_PRAC
1	jan	novák	10	studovna
2	petr	nový	15	centrála
3	jan	nováček	10	studovna

Obrázek 13 Tabulka pro ukázkou jak použít 2NF ^[18]

Za primární klíč musíme vzít tuto dvojici ČÍSLO,ČÍS_PRAC. Ale vzniknul problém. JMÉNO a PŘÍJMENÍ není závislé na primárním klíči ČÍS_PRAC. A NÁZEV_PRAC není závislý na ČÍSLO. Aby platila 2NF musíme tabulku rozdělit na dvě tabulky (PRACOVNÍK a PRACOVISTĚ). Odborně se to označuje dekompozice relačního schématu.

ČÍSLO	JMÉNO	PŘÍJMENÍ	ČÍS_PRAC
1	jan	novák	10
2	petr	nový	15
3	jan	nováček	10

ČÍSLO	NÁZEV
10	studovna
15	centrála

Obrázek 14 Tabulky podle 2NF ^[18]

3. Třetí normální forma (3NF) – Tabulka musí splňovat 2NF a každý atribut, který není primárním klíčem, není tranzitivně závislý na žádném klíči.
Pro vysvětlení pojmu tranzitivní závislost použijeme tabulku PLATY - Obrázek 15.

ČÍSLO	JMÉNO	PŘÍJMENÍ	FUNKCE	PLAT
1	jan	novák	technik	15000
2	petr	nový	vedoucí	21500
3	jan	nováček	správce	17500

Obrázek 15 Tabulka PLATY ^[18]

Tabulka sice nesplňuje 2NF, což je podmínkou pro 3NF. Ale to na vysvětlení tranzitivní závislosti nevádí. Atributy JMÉNO, PŘÍJMENÍ a FUNKCE závisí na atributu ČÍSLO. A atribut PLAT je závislý na atributu FUNKCE. Z toho vyplývá že, když ČÍSLO->FUNKCE a FUNKCE->PLAT. Tak vznikne jev tranzitivita, kdy ČÍSLO->PLAT. Na úpravu tabulky použijeme dekompozici a vzniknou nám tabulky FUNKCE a PLATY.

ČÍSLO	JMÉNO	PŘÍJMENÍ	FUNKCE
1	jan	novák	technik
2	petr	nový	vedoucí
3	jan	nováček	správce

FUNKCE	PLAT
technik	21500
vedoucí	17500
správce	15000

Obrázek 16 Tabulky podle 3NF ^[18]

Podle základních tří normálních forem jsou tabulky v Obrázek 16 v pořádku. Ale z praktického hlediska je lepší když každá tabulka má svůj jednoduchý primární klíč.

ČÍSLO	JMÉNO	PŘÍJMENÍ	CIS_FUN
1	jan	novák	121
2	petr	nový	156
3	jan	nováček	127

ČÍSLO	FUNKCE	PLAT
121	technik	21500
156	vedoucí	17500
127	správce	15000

Obrázek 17 Tabulky splňující všechny normální formy ^[18]

4. Boyce-Coddova normální forma (BCNF) – Pro dvě množiny atributu A a B platí A->B. A množina B není podmnožinou A. Pak A obsahuje primární klíč. BCNF zjednodušuje práci s tabulkami a platí, když jsou splněny 1NF, 2NF a 3NF. Tak je i splněna BCNF. ^[18]

4.2 Jazyk SQL

Vznik jazyka SQL je kolem 70. a 80. let. V roce 1986 byl uznán první standard, označený jako SQL86. Další verze vznikla v roce 1992 a řešila nedostatky předchozí verze. Je označována jako SQL92 a používá se dodnes. Zkratka SQL znamená Structured Query Language. Jazyk v sobě obsahuje nástroje pro tvorbu a úpravu dat databází.

SQL je tzv. deklarativní programovací jazyk. Což znamená, že kód jazyka SQL pro jeho funkčnost musíme vložit do jiného procedurálního programovacího jazyka. Jako je například jazyk C. Samotný jazyk SQL se dá používat jen v příkazovém řádku na SQL serveru.

Jazyk SQL se skládá z několika částí. Každá část je určena pro administrátory a návrháře databázových systémů, jiné pak pro koncové uživatele a programátory.

1. DDL - Data Definition Language pomocí něhož se vytváří databázové schémata a katalogy.
2. SDL - Storage Definition Language definuje způsob uložení tabulek.
3. VDL - View Definition Language tato část je pro návrháře a správce. Vytváří pohledy, které si lze představit jako virtuální tabulku složenou z jiných tabulek.
4. DML - Data Manipulation Language je určený převážně pro koncové uživatele a programátory databázových aplikací. Obsahuje příkazy jako INSERT, UPDATE, DELETE a nejpoužívanější příkaz SELECT. ^[18]

Příklad vytvoření tabulky databáze pomocí jazyka SQL:

```
CREATE TABLE IF NOT EXISTS `ridic` (  
  `ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Prijmeni` varchar(10) COLLATE utf8_czech_ci NOT NULL,  
  `Jmeno` varchar(10) COLLATE utf8_czech_ci NOT NULL,  
  `Cislo_RP` varchar(10) COLLATE utf8_czech_ci NOT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci AUTO_INCREMENT=2 ;
```

Obrázek 18 Jazyk SQL

Na Obrázek 18 můžete vidět vytvoření tabulky řidič s atributy ID, Prijmeni, Jmeno, Cislo_RP.

Atribut ID obsahuje primární klíč. Za názvem je datový typ atributu. A v závorce je maximální počet míst jaký je možno zapsat do daného atributu.

Jazyk SQL podporuje tyto datové typy:

CHARACTER, CHARACTER VARYING, BIT, BIT VARYING, NUMERIC, DECIMAL, INTEGER, SMALLINT, FLOAT, REAL, DOUBLE PRECISION, DATE, TIME, TIMESTAMP, INTERVAL

5 Vlastní návrh aplikace

V navržené aplikaci se budou zobrazovat základní údaje z automobilu. Aplikace bude číst data z navržené databáze. Náplní bakalářské práce nebylo získávání dat, ale vytvoření databáze pro data. A následné zpracování a zobrazení. Důležitý prvek je, v jakém časovém intervalu budou data měřena / ukládána. Pro některé hodnoty je zbytečné, aby byly ukládány každou vteřinou, jako je vnější teplota. Tato hodnota se nijak razantně nemění. Ale třeba údaje o poloze se při rychlosti 120 km/h za 2 min dokážou změnit až o 4 km. Proto je důležité, aby zvolený čas byl optimální pro všechny měřené parametry.

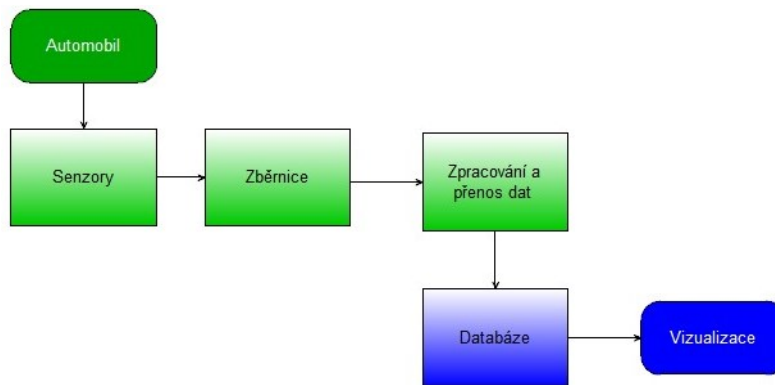
Grafické rozhraní je rozvrhnuté pro rychlou orientaci a komfort uživatele. Je zde použito několik jednoduchých grafických prvků pro lepší funkčnost a jednoduchou rozšiřitelnost aplikace.

Jak již bylo zmíněno, je potřeba, aby některé veličiny byly měřeny častěji než jiné. Nyní si popíšeme některé měření podrobněji. Nebylo by vhodné každou minutu ukládat a aktualizovat data, které se mění každou půlminutu. Z toho důvodu je rozumné měřit zrychlení každou vteřinu a polohu půlminutu. A teplotu každých 10 minut, abychom měli dostatečné množství údajů o jízdě vozidla.

Dále je potřeba správně nastavit databázi a to tak, aby se do ní neukládalo zbytečné množství dat. Protože místo na serveru není nekonečně veliké. Z tohoto důvodu se některé údaje zpracovávají přímo v aplikaci. Jako například průměrná rychlost vozidla a doba jízdy.

5.1 Měřicí a vizualizační řetězec

Měřicí část se skládá ze tří částí. První část jsou senzory ve vozidle. Druhá je sběrnice, která nám umožní přenos dat od senzoru k dalšímu zpracování. Poslední část je zpracování a přenos dat. Vizualizační část se skládá ze dvou částí. První část je databáze, kde jsou uchovávány naměřená data. A druhá část je samotná vizualizace.



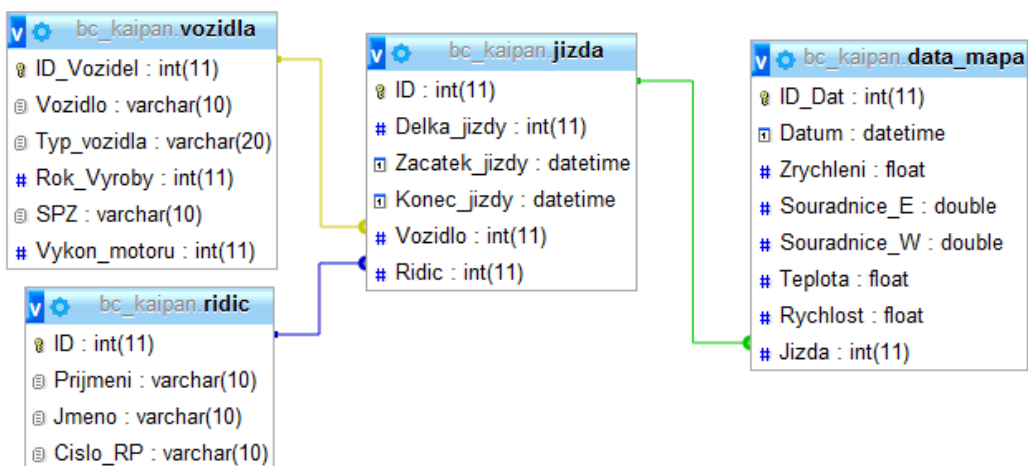
Obrázek 19 Měřicí a vizualizační řetězec

6 Softwarové řešení

Tato kapitola se zabývá databází a výslednou aplikací. Je zde popsán základní princip aplikace, co aplikace obsahuje i části zdrojového kódu.

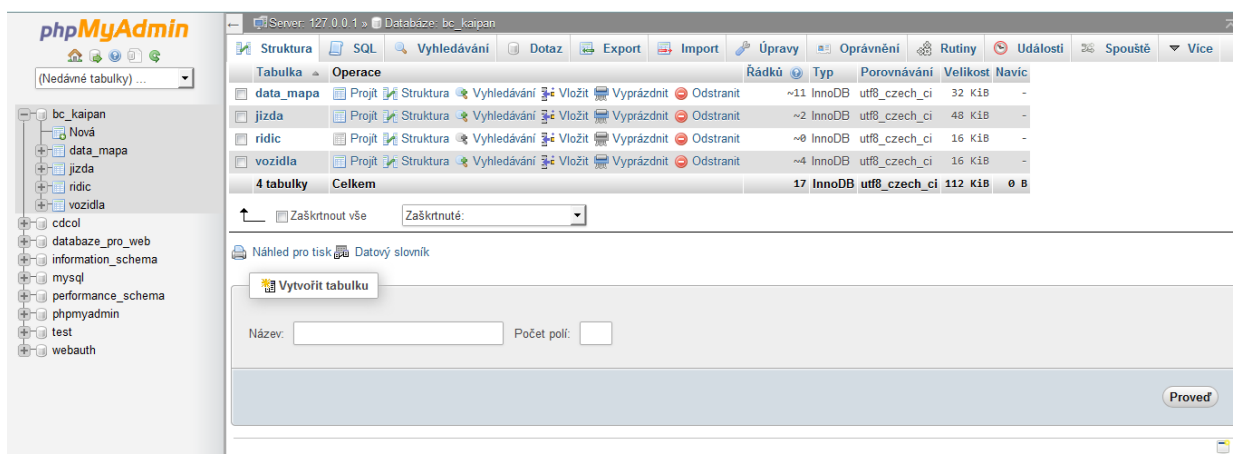
6.1 Struktura databáze

Nejtěžší byl návrh databáze, aby splňovala všechny normální formy. A aby byla jednoduše rozšiřitelná o další parametry. Databáze *bc_kaipan* se skládá ze čtyř tabulek a bude evidovat **vozidlo** (**ID_Vozidel** – to je identifikační číslo vozidla, **Vozidlo** – označení vozidla např.: Vozidlo1 z důvodu snadnějšího rozpoznání vozidla, **Typ_vozidla**- zde je uvedeno značka a typ vozidla, **Rok_Vyroby**- rok výroby vozidla, **SPZ**-státní poznávací značka vozidla, **Vykon_motoru** – výkon motoru v kW). Dále také řidiče, který dané vozidlo řídil v **ridic** (**ID** – identifikační číslo řidiče, **Prijmeni**, **Jmeno**, **Cislo_RP** – číslo řidičského průkazu). Pak je zde **jizda** (**ID**- identifikační číslo jízdy, **Delka_jizdy** – vzdálenost ujetá od startu jízdy do cíle, **Zacatek_jizdy** – čas kdy začala jízda, **Konec_jizdy** – čas kdy jízda skončila, **Vozidlo** – ID vozidla které danou jízdu jelo, **Ridic** – ID řidiče který danou jízdu jel) a nakonec **data_mapa** (**ID_Dat** – identifikační číslo dat, **Datum** – časový údaj kdy data byla pořízena, **Zrychleni** – zrychlení v daném bodě měření, **Souradnice_E** a **Souradnice_W** jsou to GPS souřadnice o poloze vozidla, **Teplota**- vnější teplota, **Rychlost**, **Jizda** – ID jízdy slouží k určení které jízdě patří dané údaje)



Obrázek 20 Databáze použitá pro aplikaci

Databáze je typu MySQL. A byla tvořena pomocí webového rozhraní phpMyAdmin. Tento nástroj je napsán v jazyce PHP. Díky tomu, že využívá webové rozhraní, umožňuje tak jednoduchou správu databází MySQL. Ke komunikaci s databází se dá použít jazyk SQL. Účel phpMyAdmin je umožnit kompletní správu serverů a dat MySQL.



Obrázek 21 Ukázka webového rozhraní phpMyAdmin

Základní výbavou phpMyAdmin je:

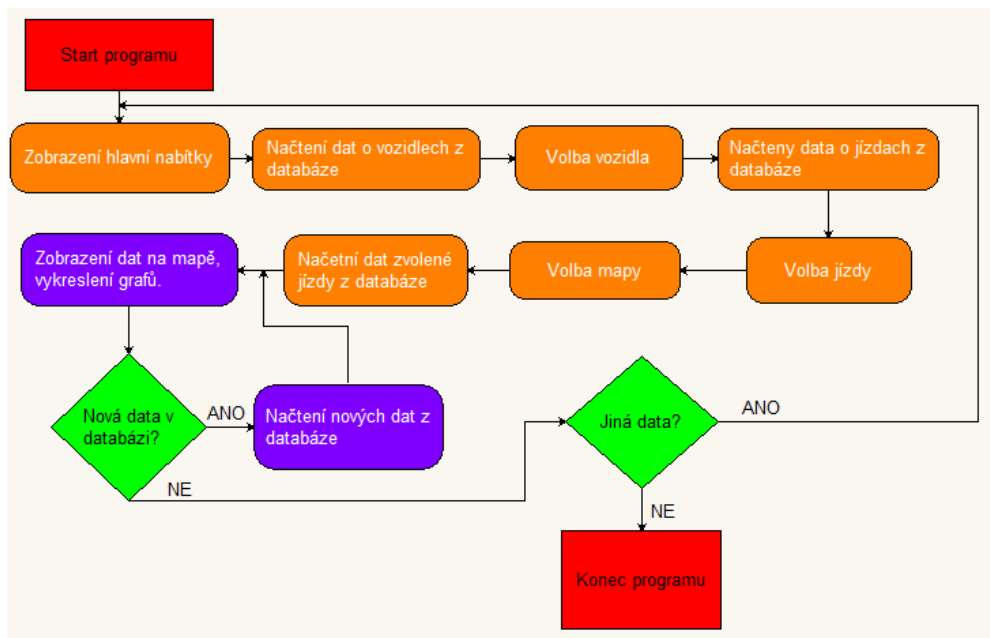
- Vytvoření a odstranění databáze
- Vytvoření, odstranění, kopírování, přejmenování tabulky
- Vyhledávání dat v databázi pomocí SQL dotazu
- Možnost více uživatelů a jejich oprávnění
- Cizí klíče
- Kontrola databázových oprávnění
- A mnoho dalších

Výhody databáze MySQL:

- Stabilita
- Vysoká rychlost databáze
- MySQL je portováno na většinu operačních systémů
- Umožňuje přístup z mnoha programovacích jazyků
- Standardní verze je zdarma
- Velmi rozšířená

6.2 Vizualizační aplikace

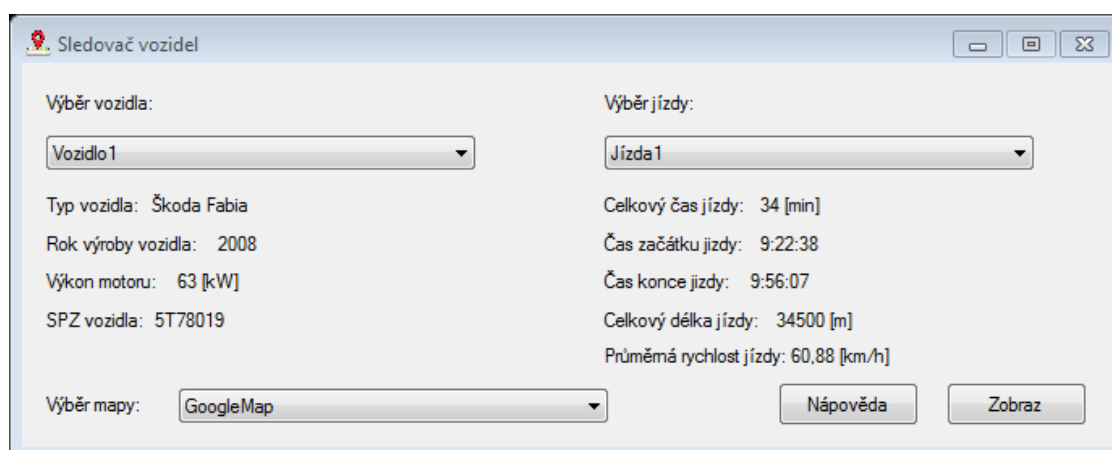
Aplikace je tvořena aby měla intuitivní ovládání a data v ní zobrazena byla jednoduše čitelná.



Obrázek 22 Vývojový diagram aplikace

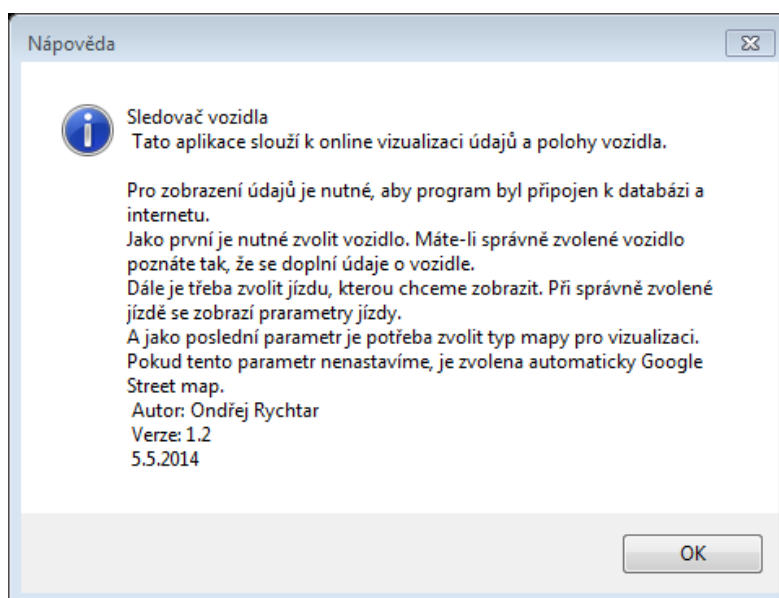
Na Obrázek 22 můžete vidět zjednodušený vývojový diagram aplikace. V aplikaci se nachází více rozhodovacích úrovní a několik výpočetních operací. Je tam například volba barvy trasy podle rychlosti vozidla a výpočet doby trasy. Ale na tyto části je zaměřeno později.

Start aplikace je kliknutím na ikonu. Při startu aplikace se zobrazí okno hlavní nabídky. A zároveň se načtou data o vozidlech z databáze. Jako třeba typ, rok výroby, výkon motoru a SPZ vozidla. Poté co uživatel zvolí vozidlo z nabídky. Jsou načteny data o jízdách zvoleného vozidla z databáze. Dále je vypočtena doba jízdy a průměrná rychlost vozidla a následně zobrazena v aplikaci. Po zvolení jízdy a typu mapy jsou po stisknutí tlačítka *Zobrazit* načtena a zobrazena data na mapě, v grafu a dalších grafických prvcích. Od této chvíle se spustí časovač, který po určitém časovém intervalu ověří zda, v databázi k vybrané jízdě nepříbyla nová data. Pokud ano jsou nová data vykreslena ke stávajícím. Pokud ne, uživatel si může dále prohlížet zobrazená data, vrátit se do hlavní nabídky a změnit vozidlo, jízdu nebo mapu, na které chce daná data zobrazit. A celý proces se opakuje znovu. Aplikace se dá ukončit v libovolné části procesu stiskem křížku.



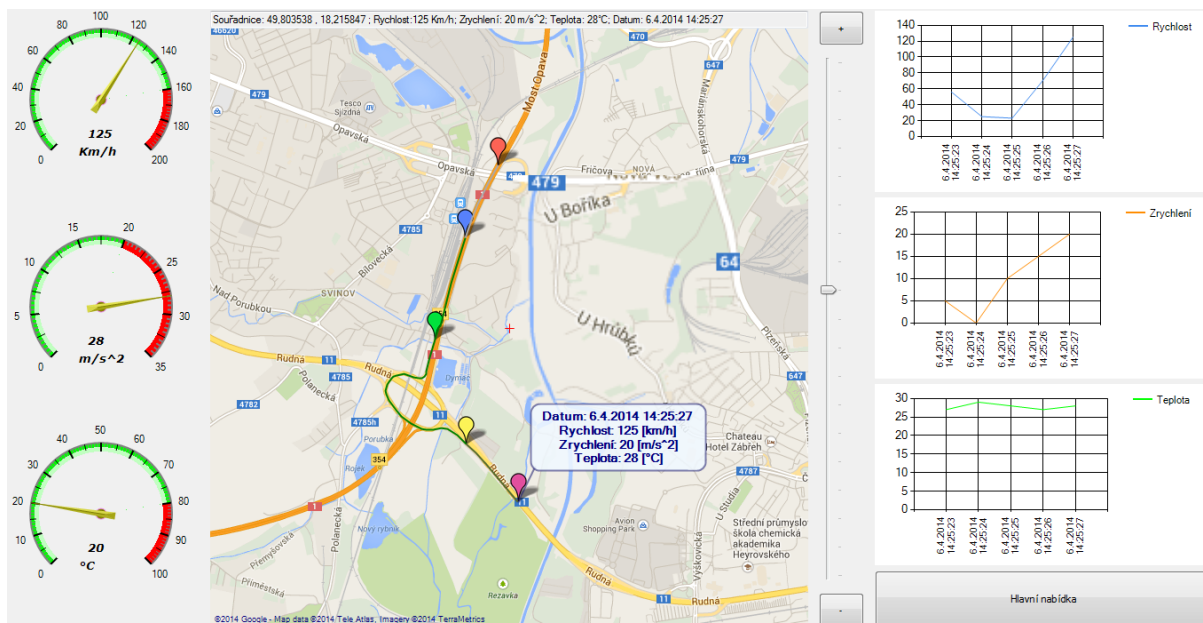
Obrázek 23 Základní okno aplikace

Na Obrázek 23 vidíte základní okno aplikace. Nachází se zde nabídka pro výběr vozidla. Při výběru vozidla se doplní informace o typu, roku výroby, výkonu motoru a SPZ vozidla. Jak můžete vidět na obrázku je již zvoleno *Vozidlo1* a informace o vozidle doplněny. Dále se zobrazí jízdy v menu *Výběr jízdy*. Při volbě jízdy se pod menu zobrazují informace o dané jízdě. Celkový čas jízdy, čas na začátku a konci jízdy, ujetá vzdálenost. Dále je zde menu pro výběr map. Jsou zde mapy ze serveru Google a Mapy.cz. Od Google jsou zde GoogleMap, GoogleSatelliteMap, GoogleHybridMap a od Mapy.cz CzechMap, CzechSatelliteMap, CzechHybridMap a CzechTuristMap. Mapy se dají zvolit kdykoliv. To znamená, že není potřeba zvolit vozidlo nebo jízdu. Je zde také tlačítko pro nápovědu, kde jsou základní informace a nápověda pro ovládání aplikace Obrázek 24.



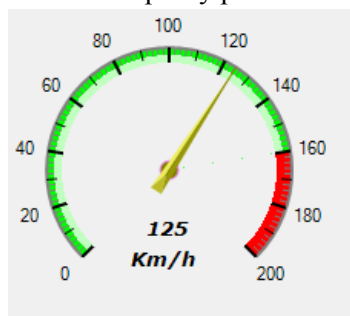
Obrázek 24 Okno nápovědy

Nachází se zde tlačítko *Zobraz*, které slouží pro zobrazení zvolených údajů a zobrazení okna s mapou. Pokud není zvolené vozidlo a jízda, na mapě se nezobrazí žádné body, trasy. U okna je zablokované tlačítko pro maximalizaci, protože okno má optimální velikost a není potřeba jí měnit.



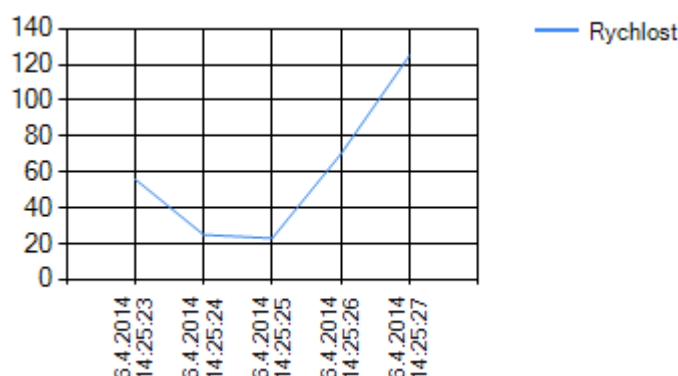
Obrázek 25 Okno aplikace pro vizualizaci

Na Obrázek 25 lze vidět okno pro vizualizaci dat. Obsahuje několik zobrazovacích prvků pro lepší přehlednost údajů. Jako hlavní prvek je mapa, kde se nachází body a trasa mezi body. Každý bod má barvu podle zrychlení v daném měřeném bodě. A navíc při najetí myši na bod se zobrazí bublina z měřených dat v daném bodě. Dále trasa mezi body se vybarví podle rychlosti změřené v bodě, ze kterého vychází. Vpravo od mapy se nachází tři prvky pro zoom mapy. Jeden posuvník a dvě tlačítka.



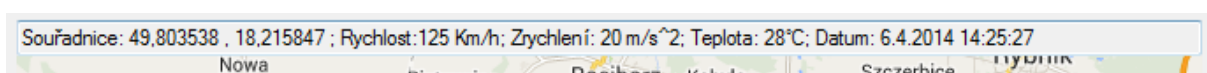
Obrázek 26 Analogový ukazatel rychlosti

V levé části můžete vidět analogový ukazatel, který je na Obrázek 26. Pro přehlednost jsou data zobrazena jako v automobilu. Pomocí analogového ukazatele jsou zobrazovány rychlost, zrychlení a vnější teplota. Na ukazatelích jsou vždy zobrazeny poslední hodnoty.



Obrázek 27 Zobrazení grafu v aplikaci

V pravé části se nacházejí tři grafy, jaký například vidíte na Obrázek 27. V grafech je zobrazena rychlost, zrychlení a vnější teplota s časovými údaji. Pro lepší srovnání jsou data zobrazena v grafech. Nová data se vždy přiřazují nakonec grafu. Každá veličina je zobrazena jinou barvou. Rychlost je modře, zrychlení žlutě a teplota zeleně.



Obrázek 28 Stavový řádek

Také je zde stavový řádek, který se nachází v horní části aplikace nad mapou. Ve stavovém řádku je uvedeno, kde se dané vozidlo nacházelo naposledy, jakou rychlostí jelo. Jaké mělo zrychlení v daném místě. Při jaké venkovní teplotě a kdy.

Poslední prvek, který se nachází v okně, je tlačítko, pomocí něhož se vrací zpět do hlavní nabídky.

Data jsou čtena z databáze pomocí nástroje Connector/Net. Tento nástroj je určený pro propojení databáze MySQL s C# aplikací. Je volně ke stažení na webových stránkách tvůrců typu databáze MySQL. Connector funguje následovně. V třídě DBConnect je vytvořen inicializační textový řetězec pro připojení k databázi - Obrázek 29. Pro připojení je potřeba nastavit několik parametrů, jako je třeba server, na kterém je databáze umístěna. Dále název databáze, uživatelský účet a heslo. Bez těchto parametrů není možné se k databázi připojit.

```
// Inicializace hodnot
private void Initialize()
{
    server = "rc112.vsb.cz";
    port = "3306";
    database = "bc_kaipan";
    uid = "bc_kaipan";
    password = "ZUZA6XPheqfwNPRv";
    string connectionString;
    connectionString = "SERVER=" + server + ";" + "PORT=" + port + ";" + "DATABASE=" +
        database + ";" + "UID=" + uid + ";" + "PASSWORD=" + password + ";";

    connection = new MySqlConnection(connectionString);
}
```

Obrázek 29 Inicializační textový řetězec

Aplikace se pomocí textového řetězce pokusí navázat spojení s databází. Pokud je spojení v pořádku, je umožněno vyslání SQL dotazu, ale pokud ne, objeví se chybová hláška. Nejčastější jsou tyto dvě chyby. Za prvé nelze se připojit k serveru a za druhé je neplatné uživatelské heslo nebo jméno. Pokud to není ani jedna z předchozích chyb, je zde možnost pro zobrazení chybové hlášky od Connectoru. Na Obrázek 30 můžete vidět tuto část kódu.

```
// Otevření připojení k databázi
private bool OpenConnection()
{
    try
    {
        connection.Open();
        return true;
    }
    catch (MySqlException ex)
    {
        // Při manipulaci chyby, můžete na základě odezvy vaší aplikace
        // Na číslo chyby.
        // Dva nejběžnější čísla chyb při připojování jsou následující:
        // 0: Nelze se připojit k serveru.
        // 1045: Neplatné uživatelské jméno a / nebo heslo.
        switch (ex.Number)
        {
            case 0:
                MessageBox.Show("Nelze se připojit k serveru");
                break;

            case 1045:
                MessageBox.Show("Neplatné uživatelské jméno / heslo.");
                break;

            default:
                MessageBox.Show(ex.ToString());
                break;
        }
        return false;
    }
}
```

Obrázek 30 Otevření spojení s databází

Po úspěšném navázání spojení je vyslán SQL dotaz. Odpovědi SQL dotazu jsou data ve formě string, která jsou uloženy do struktury List. Pro správné uložení je potřeba vědět název odpovědi na SQL dotaz. Poté je spojení uzavřeno - Obrázek 31.

```

string query = "SELECT * FROM vozidla";

//Create a list to store the result
List<string>[] list = new List<string>[6];
list[0] = new List<string>();
list[1] = new List<string>();
list[2] = new List<string>();
list[3] = new List<string>();
list[4] = new List<string>();
list[5] = new List<string>();

//Open connection
if (this.OpenConnection() == true)
{
    //Create Command
    MySqlCommand cmd = new MySqlCommand(query, connection);
    //Create a data reader and Execute the command
    MySqlDataReader dataReader = cmd.ExecuteReader();

    //Read the data and store them in the list

    while (dataReader.Read())
    {
        list[0].Add(dataReader["ID_Vozidel"] + "");
        list[1].Add(dataReader["Vozidlo"] + "");
        list[2].Add(dataReader["Typ_vozidla"] + "");
        list[3].Add(dataReader["Rok_Vyroby"] + "");
        list[4].Add(dataReader["SPZ"] + "");
        list[5].Add(dataReader["Vykon_motoru"] + "");
    }

    //close Data Reader
    dataReader.Close();

    //close Connection
    this.CloseConnection();

    //return list to be displayed
    return list;
}

```

Obrázek 31 Čtení dat z databáze

Ze struktury List jsou data převedena do formátu, v jakých jsou dále používána. Jako například údaje o vozidle jsou zobrazeny v okně hlavní nabídky.

Další důležitý prvek aplikace je mapa. Pro mapu je zde použita multiplatformní komponenta GMap.NET. Tato komponenta umožňuje tvorbu bodů, tras a polygonů.

Body jsou tvořeny na souřadnicích získaných z databáze, barvu mají podle zrychlení v daném bodě. Dále je bodu přiřazena bublina s daty - Obrázek 32. Je možné volit z několika druhů bodů. V aplikaci byl použit základní typ od Google.

```

GMapOverlay markersOverlay = new GMapOverlay("markers");
_form.Gmap.Zoom = 10;
if (_zrychleni[i]==0)
{
    GMarkerGoogle marker = new GMarkerGoogle(new PointLatLng(_souradnice_E[i], _souradnice_W[i]), GMarkerGoogleType.blue);
    _form.Gmap.Position = new PointLatLng(_souradnice_E[i], _souradnice_W[i]);
    marker.ToolTip = new GMapRoundedToolTip(marker);
    marker.ToolTipText = "Datum: " + _datum[i] + "\n" + "Rychlost: " + _rychlost[i].ToString() + " [km/h] \n" + "Zrychlení: " +
    markersOverlay.Markers.Add(marker);

    _form.Gmap.Overlays.Add(markersOverlay);
}

```

Obrázek 32 Tvorba bodu

Trasy jsou tvořeny mezi dvěma body a mají barvu podle rychlosti v prvním bodě trasy - Obrázek 33. Souřadnice jsou uloženy v poli pro lepší práci s daty a jsou čteny pomocí funkce for.

Barvu trasy lze zvolit z mnoha barev. Například při nulové rychlosti má trasa červenou barvu. Barvy jsou zvolené tak, aby se neztrácely v barvách mapy. Nebo při zvolení hybridní mapy v čarách, které znázorňují silnici. Také lze zvolit tloušťku čáry, která zde byla zvolena 2. Při větší tloušťce čáry se může stát, že vykreslená trasa překryje vozovku nebo naopak, při příliš tenké trase zanikne v mapě.

```
for (int k = 0, j = 1; j < _souradnice_E.Length; k++, j++)
{
    PointLatLng start = new PointLatLng(_souradnice_E[k], _souradnice_W[k]);
    PointLatLng end = new PointLatLng(_souradnice_E[j], _souradnice_W[j]);

    MapRoute route = GMap.NET.MapProviders.GoogleMapProvider.Instance.GetRoute(
        start, end, false, false, 15);
    GMapRoute r = new GMapRoute(route.Points, "My route");
    GMapOverlay routesOverlay = new GMapOverlay("routes");
    routesOverlay.Routes.Add(r);
    _form.Gmap.Overlays.Add(routesOverlay);
}

if (_rychlost[k] == 0)
{
    Pen pen = new Pen(Color.Red);
    pen.Width = 2;
    r.Stroke = pen;
}
```

Obrázek 33 Tvorba trasy

Další prvky vizualizace jsou grafy. Nahrání dat do grafu je jednoduché. Data jsou přidávána jako body za sebou a jsou propojeny - Obrázek 34. Každá veličina má svůj vlastní graf. Grafy dále umožňují umístit několik křivek do jednoho grafu. Dále lze zvolit barvu, typ bodu a tloušťku čáry. Také je zde možnost zvolit libovolný typ grafu. Jako například graf sloupcový nebo koláčový. Jak bylo uvedeno výše, byl zvolen graf bodový. Pro lepší zobrazení naměřených hodnot a jednodušší srovnání mezi sebou.

```
_form.chart1.Series["Rychlost"].Points.AddXY(_datum[x], _rychlost[x]);
_form.chart2.Series["Zrychlení"].Points.AddXY(_datum[x], _zrychlení[x]);
_form.chart3.Series["Teplota"].Points.AddXY(_datum[x], _teplota[x]);
```

Obrázek 34 Tvorba grafů

Jeden z hlavních prvků vizualizace jsou analogové ukazatele - Obrázek 26. Pomocí kterých je zobrazována teplota, rychlost a zrychlení. Analogové ukazatele ukazují poslední známou hodnotu z jízdy - Obrázek 35. U každého z analogových ukazatelů je nastavena minimální a maximální hodnota tak, aby zvolený rozsah byl, co nejbližší reálným hodnotám. U ukazatele zrychlení se počítá s tím, že hodnota zrychlení bude kladná. A to i při brždění vozidla bude převedeno na kladnou hodnotu.

```
_form.aGauge1.Value = Convert.ToSingle(_rychlost[_rychlost.Length-1]);
_form.aGauge2.Value = Convert.ToSingle(_zrychlení[_zrychlení.Length - 1]);
_form.aGauge3.Value = Convert.ToSingle(_teplota[_teplota.Length - 1]);
```

Obrázek 35 Nastavování analogových ukazatelů

Další prvek je stavový řádek - Obrázek 28. Je to textbox, do něhož jsou data převedena na string a zobrazena.

Aby aplikace fungovala v režimu online. Je v aplikaci umístěn časovač, který po stanoveném časovém intervalu vyšle dotaz, zda v databázi nepříbyla data. A pokud ano, tak jsou data přidána - Obrázek 36.

```
private void timer1_Tick(object sender, EventArgs e)
{
    this.Text = "Mapa " + ++this.time + "s";

    if (time == 60)
    {
        newID = Convert.ToInt32(connect.Select_newData_mapa(ID));
        if (newID != lastID)
        {
            Tvorba tvorba = new Tvorba(this, ID);
        }
        time = 0;
    }
}
```

Obrázek 36 Režim online

Při zobrazování údajů o jízdě je proveden jednoduchý výpočet doby trasy. Tento výpočet se provádí, aby nebyla zbytečně zatěžována databáze - Obrázek 37. A to tak, že se od koncového času, odečte počáteční čas. Čas je vypočten v minutách.

```
public int Celkovy_cas()
{
    int Cas, h,m;
    m = Cas_konec.Minute - Cas_start.Minute;
    h = Cas_konec.Hour - Cas_start.Hour;
    Cas = m+ (60*h);
    return Cas;
}
```

Obrázek 37 Výpočet doby jízdy

Další výpočet v aplikaci je průměrná rychlost vozidla. Je vypočtena z celkové doby a délky jízdy vozidla - Obrázek 38. Doba jízdy musí být převedena z integeru na double a následně vynásobena šedesáti pro převod z minut na sekundy. Dále celý výpočet je násoben konstantou 3,6 pro převod z m/s na km/h.

```
public double Prum_rychlost(int Cas, double Delka)
{
    double Rychlost,Konst = 3.6;

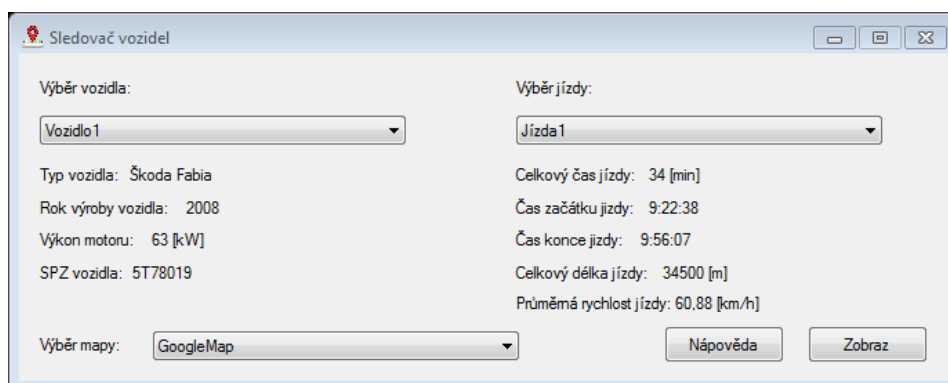
    Rychlost = (Delka/(Convert.ToDouble(Cas)*60))*Konst;

    return Rychlost;
}
```

Obrázek 38 Výpočet průměrné rychlosti vozidla

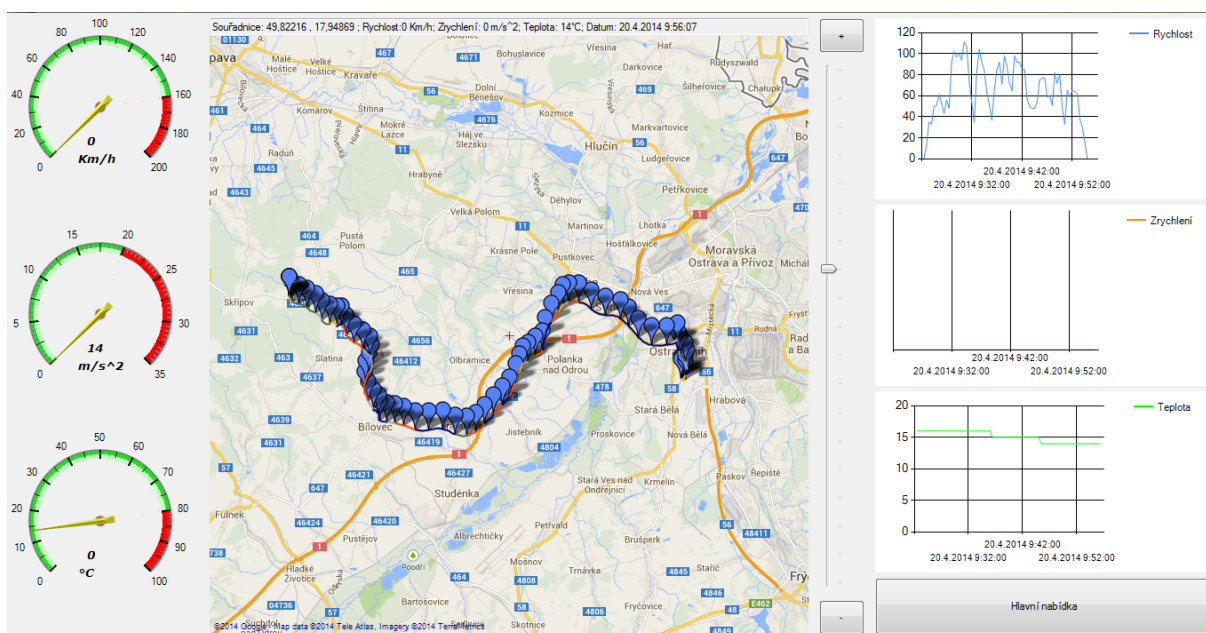
7 Testování aplikace

Testování aplikace bylo provedeno experimentálním řešením. Byla použita mobilní aplikace Vezma, naistalována na mobilním telefonu, obsahující kombinovaný systém GPS a GLONASS pro přesnější určení polohy. Tato aplikace se chová jako elektronická kniha jízd. Zaznamenává údaje o poloze, rychlosti a nadmořské výšce vozidla. Záznam o poloze provádí aplikace Vezma, co 30 s. Dále umožňuje získané údaje vyexportovat do souboru GPX nebo KML. Pro sběr dat bylo použito vozidlo Škoda Fabie. Toto vozidlo bylo také zavedeno do databáze. Data o jízdě byla vložena do databáze manuálně. Protože aplikace nezaznamenává zrychlení a teplotu, bylo zrychlení uvedeno jako nulové a teplota byla odečtena z palubního displeje.



Obrázek 39 Hlavní nabídka

Na Obrázek 39 můžete vidět okno hlavní nabídky se zvoleným testovacím vozidlem a testovací jízdou. Jízda trvala 34 minut a bylo ujetu 34,5 km průměrnou rychlostí 60,88 km/h.



Obrázek 40 Test aplikace

Na Obrázek 40 můžete vidět vizualizaci testovací jízdy. Body na mapě mají modrou barvu, což znamená, že vozidlo mělo nulové zrychlení. A to z důvodu, že zrychlení nebylo měřeno. Ověřil jsem, že trasa zobrazená na mapě je skutečná trasa, po které jelo vozidlo. A také si lze povšimnu poklesu teploty. Ten byl způsoben opuštěním městské zástavby a vyjetím vozidla do vyšší nadmořské výšky. Několik počátečních a koncových bodů má rychlost nulovou, je díky tomu, že spuštění a ukončení jízdy bylo provedeno manuálně. Před a po skončení jízdy vozidla.



Obrázek 41 Chyba souřadnic

Jak si můžete všimnout na Obrázek 41, některé body se nenacházejí v trase. Takové body byly zkontrolovány, zda nedošla chyba při manuálním zadávání souřadnic. Když jsem vyloučil tuto chybu, došel jsem k závěru, že chyba může být způsobena dvěma parametry. První je, že získané souřadnice nebyly, až tak přesné. Mohlo to být způsobeno aplikací Vezma nebo mobilním přístrojem. A druhý je, že samotné GMap.NET se snaží, umístit trasu vždy na vozovku v mapě. A díky tomu vzniká tato chyba. Zvolená mapa je Google Street.

8 Závěr

Cílem této práce bylo vytvořit aplikaci pro online vizualizaci údajů a polohy vozidla. Data měla být zobrazována na mapovém podkladu. Také měla být vytvořena databáze, ze které bude aplikace číst údaje, pro jednoduché zpracování dat. Řešení bylo experimentálně ověřeno.

První částí bakalářské práce byla tvorba databáze. V databázi jsou uložena všechna data potřebná pro následnou vizualizaci. Pro tyto účely byla použita databáze typu MySQL. Tento typ databáze má jednoduchou správu a mnoho dalších výhod. Jednou z hlavních výhod je velká rychlost databáze, která je potřeba při práci s velkým počtem dat v režimu online. Databáze je navržena tak aby byla univerzální a dala se použít pro jiné aplikace, které mohou čerpat data z již existující databáze. Pro další provoz databáze je potřeba vytvořit program, který bude promazávat starší data v databázi. Protože některé údaje jako je například zrychlení je potřeba často měřit. Tím se snadno může naplnit databáze a nezbylo by místo pro nová data. Pro tvorbu a následné testování aplikace nebylo potřeba tolik dat, aby bylo nutné vytvořit program pro automatické mazání dat. Mazání starých dat bylo provedeno manuálně.

Když byla vytvořena databáze. Bylo nutné použít funkci Connector pro propojení databáze s C# aplikací. Tato funkce je poskytována tvůrci databázi MySQL. Jejím úkolem je při připojení programu, přenést SQL dotaz do databáze a následně vrátit odpověď programu. Připojení k databázi provádí pomocí předem nadefinovaných parametrů. Jako je například IP adresa, port serveru nebo jméno a heslo uživatele. Tyto parametry lze jednoduše změnit podle potřeby. Protože funkce používá SQL dotazy lze provádět různé operace s databází. Jako například mazání nebo tvorba tabulek. Takže pomocí funkce Connector může při spuštění aplikace si sama vytvořit databázi, pokud by to bylo potřeba. Také by šlo realizovat automatické mazání starých dat.

K vizualizaci dat, byla vytvořena aplikace, která používá různé mapové podklady ze dvou GIS systémů. Oba systémy jsou určeny pro volné užívání. Jeden je od firmy Google a druhý od české firmy Mapy.cz. Od Google je možno zvolit ze tří druhů map a od Mapy.cz ze čtyř. Aplikace je napsaná v programovacím jazyku C#. Pro vložení map je použito komponenty GMap.NET. Je to volně šiřitelná komponenta pro C# aplikace. Umožňuje použití různých GIS systémů. Jako další vizualizační prvky byly použity grafy, analogové ukazatele a stavový řádek. Grafy bylo nutné doinstalovat, protože nejsou v základních komponentách okenních form aplikací. Byla použita volně dostupná komponenta Chart. Dalším prvkem jsou analogové ukazatele. Je to komponenta vytvořená pro C# a je také volně dostupná. Poslední prvek je stavový řádek. Ten byl vytvořen pro lepší přehlednost dat a o aktuální poloze vozidla. V aplikaci se na mapě zobrazuje jízda zvoleného vozidla. A to tak, že do mapy se vykreslí trasa ve falešných barvách podle rychlosti vozidla. V měřených místech se zobrazí na mapě bod v barvě podle velikosti zrychlení. Při najetí myši na bod se u něj zobrazí bublina s naměřenými daty. Je možné volit mezi několika vozidly a u každého vozidla se dá zvolit jízda. Trasa vozidla je rozdělená na jízdy, protože je zbytečné zaznamenávat údaje u vozidla, když někde delší dobu parkuje. Na analogových ukazatelích se ukazuje rychlost, zrychlení a vnější teplota. Tytéž údaje jsou zobrazeny v grafu pro lepší srovnání. V části vizualizace je umístěno několik prvků, pro snadnější ovládání mapy.

Tato práce byla velkým přínosem. Poskytla mi náhled do principu navigačních systémů. Dále do telemetrie vozidel a přenosu mobilních dat. Další oblastí, která byla pro mě nová a kterou jsem musel zvládnout, byla databáze a jazyk SQL. Díky této práci jsem si rozšířil znalosti v programovacím jazyce C#. A to hlavně v oblasti okenních form aplikací. Kde bylo nutné dohledat potřebné komponenty, doinstalovat je a naučit se je používat.

Přesto že byly splněny všechny body zadání bakalářské práce, je možné tuto aplikaci vylepšit a dále rozšířit. Ve vizualizační části se dají přidat prvky, které by zobrazovaly informace, jako například zda svítí světla vozidla, kolik jede pasažérů či teplotu motoru. Dále by mohl být umístěný export dat do textového nebo Excel souboru. Program je vytvořen jen pro operační systémy Windows. Dalším velkým vylepšením by bylo vytvoření aplikace pro chytré telefony. To by umožňovalo kontrolu stavu vozidla, i když nemáme přístup k počítači. Další vylepšení je, že jako jednoduché záznamové zařízení se dá použít mobilní telefon s aplikací, která by zaznamenávala polohu, rychlost a zrychlení vozidla. Tyto parametry zvládne zaznamenat v dnešní době skoro každý telefon. A aplikace by mohla měřena data ukládat do vytvořené databáze. Záznam by se dal spustit manuálně nebo automaticky po připojení mobilního telefonu pomocí Bluetooth s automobilem.

Tato bakalářská práce se dá využít elektromobily Kaipan na sledování dojezdové vzdálenosti při závislosti na rychlosti vozidla a okolní teplotě.

Literatura

- [1] Navstar GPS. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-01-12]. Dostupné z: <http://en.wikipedia.org/wiki/Navstar>
- [2] GLONASS. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-01-12]. Dostupné z: <http://en.wikipedia.org/wiki/GLONASS>
- [3] Galileo (satellite navigation). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-01-12]. Dostupné z: [http://en.wikipedia.org/wiki/Galileo_\(satellite_navigation\)](http://en.wikipedia.org/wiki/Galileo_(satellite_navigation))
- [4] Beidou Navigation Satellite System. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001, 2. prosince 2013 [cit. 2014-01-13]. Dostupné z: http://en.wikipedia.org/wiki/Beidou_Navigation_Satellite_System
- [5] GPS. [online]. [cit. 2014-01-14]. Dostupné z: <http://gps.slansko.cz/>
- [6] MERHART, Leoš a Miloš CIMBÁLNÍK. Vyšší geodézie 2. 1. vyd. Praha: ČVUT, 1997, str. 85 – 110. ISBN 80-01-01628-5.
- [7] Galileo navigace. [online]. 2010-2011 [cit. 2014-01-14]. Dostupné z: <http://www.galileonavigace.cz/>
- [8] ING. MARTIN Hrubý, Ph.D. Geografické Informační Systémy (GIS). Brno, 2006. Dostupné z: <http://perchta.fit.vutbr.cz/vyuka-gis/uploads/1/GIS-final2.1.pdf>
- [9] DOC. ING. VÁCLAV, Čada CSc. Souřadnicové systémy [online]. [cit. 2014-01-15]. Dostupné z: <http://gis.zcu.cz/studium/gen1/html/ch02s03.html>
- [10] Telemetrie. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2. ledna 2014 [cit. 2014-01-15]. Dostupné z: <http://en.wikipedia.org/wiki/Telemetry>
- [11] UTM. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 26. 5. 2013 [cit. 2014-01-15]. Dostupné z: <http://cs.wikipedia.org/wiki/UTM>
- [12] PETERKA, Jiří. Celulární (buňkový) princip. EArchív.cz. Dostupné z: <http://www.earchiv.cz/a008s200/a008s201.php3>
- [13] VELICKÝ, Tomáš. Datové přenosy po GSM sítích, technologie HSCSD, GPRS a UMTS [online]. České Budějovice, 2002 [cit. 2014-01-15]. Dostupné z: <http://www.petrpexa.cz/diplomky/velicky.pdf>
- [14] Co je to CAN BUS?. Autoarmy.cz [online]. [cit. 2014-01-18]. Dostupné z: <http://www.autoarmy.cz/clanky/co-je-to-can-bus.html>
- [15] Car Control [online]. [cit. 2014-01-18]. Dostupné z: <https://carcontrol.cz.o2.com/web/predstavenisluzby.aspx>
- [16] Geographic information system. In: Wikipedia: the free encyclopedia [online]. San Francisco

- (CA): Wikimedia Foundation, 2001- [cit. 2014-02-22]. Dostupné z: http://en.wikipedia.org/wiki/Geographic_information_system
- [17] Geographic Information Systems for Today and Beyond. Geographic Information Systems for Today and Beyond [online]. [cit. 2014-02-22]. Dostupné z: http://catalog.flatworldknowledge.com/bookhub/reader/3798?e=campbell_1.0-ch01_s03#
- [18] SKŘIVAN, Jaromír. Databáze: Databáze a jazyk SQL. In: Interval.cz [online]. 2000 [cit. 2014-03-04]. Dostupné z: <http://interval.cz/clanky/databaze-a-jazyk-sql/>
- [19] Využití akcelerometrů pro měření náklonu a vibrací. Micro [online]. [cit. 2014-03-09]. Dostupné z: <http://www.micro.feld.cvut.cz/home/X34SES/cviceni/Navody%20na%20cviceni/01.Akcelerometry.pdf>

Seznam příloh

Příloha I. Vizualizace dat na hybridních mapách od Google

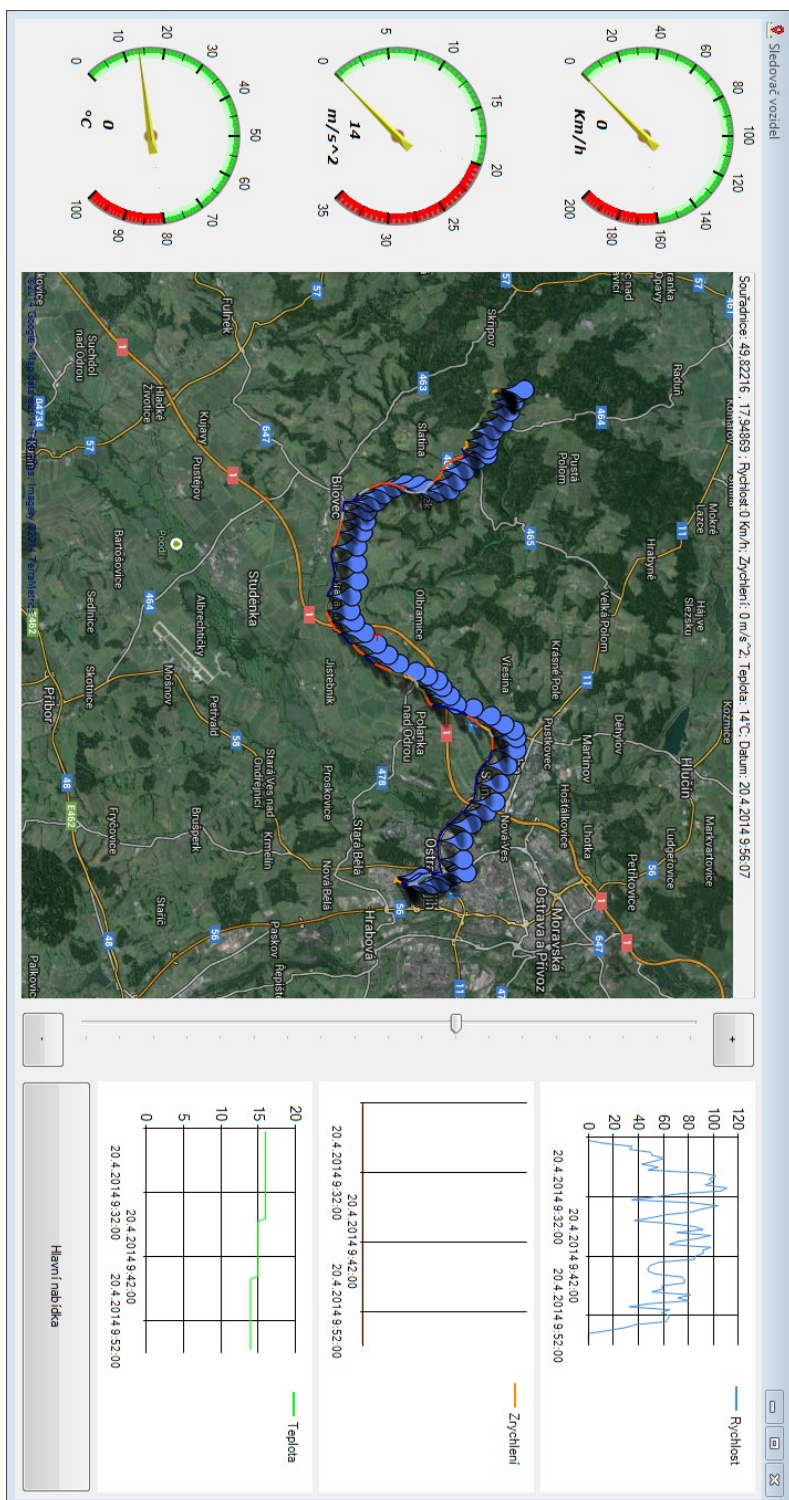
Příloha II. Vizualizace dat na mapách od Mapy.cz

Příloha III. Program Sledovač vozidla (CD nosič)

Příloha IV. Zdrojový kód Sledovače vozidla (CD nosič)

Příloha V. Databáze (CD nosič)

Příloha I. Vizualizace dat na hybridních mapách od Google



Příloha II. Vizualizace dat na mapách od Mapy.cz

